CLEARANCE OF NONLINEAR FLIGHT CONTROL LAWS USING CONTINOUS GENETIC ALGORITHMS

By
Ali Ghaleb Ali Al-Asasfeh

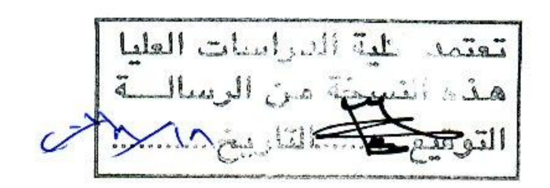
Supervisor
Dr. Mohammad Nader Jamil Hamdan, Prof.

Co-Supervisor
Dr. Zaer Salem Abo Hammour

This Dissertation was Submitted in Partial Fulfillment of the Requirements for the Doctor of Philosophy Degree in Mechanical Engineering

Faculty of Graduate Studies
University of Jordan

August 2011



COMMITTEE DECISION

This Thesis/Dissertation (*Clearance of Nonlinear Flight Control Laws Using Continuous Genetic Algorithms*) was Successfully Defended and Approved on 10th August 2011

Examination Committee	**Signature**
Dr. Mohammed Nader J. Hamdan , (Supervisor) Professor.	
Dr. Za'er Salem Abo-Hammour. (Co-Supervisor) Assistance Professor	------------------------------
Dr. Saad M. Habali (Member) Professor	 ------------------------------
Dr. Khaldoun K. Tahboub (Member) Associate Professor	------------------------------
Dr. Mahmoud A. Barghash (Member) Assistance Professor	------------------------------
Dr. Mohammad A. Alia (Member) Associate Professor (Al-Balqa' Applied University)	------------------------------



DEDICATION

TO

MY FAMILY……

MY FRIENDS……

AKNOWLEDGEMENT

I would like to express my sincerest gratitude to my advisors, Prof. Mohammed Nader J. Hamdan and Dr. Za`er Abo-Hammour, for giving me the opportunity to work under their supervision. They opened the horizon for me to get-inside the evolutionary technique and the nonlinear flight control dynamics. Their excellences in the field and enthusiasm to work extended hours have always been a source of inspiration. Special thanks also to the Faculty of Engineering, Mechanical Department staff.

Finally, I want to thank my family, *Mother*, *father*, and my *wife*, for everything they have done for me.

The dream of yesterday…is the hope of today…and the reality of tomorrow.
Robert H. Goddard, 1904

Table of Contents

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS OR SYMBOLS

List of Symbols

a	speed of sound (m/s)
bref	Wingspan (m)
cref	Mean aerodynamic chord (m)
Cl	Coefficient of rolling moment
Clβ	Rolling moment coefficient derivative with respect to β
Cm	Coefficient of pitching moment
Cmα	Pitching moment coefficient derivative with respect to α
Cmq	Pitching moment coefficient derivative with respect to q
Cn	Coefficient of yawing moment
Cnβ	Yawing moment coefficient derivative with respect to β
Cnr	Yawing moment coefficient derivative with respect to r
C_D	Coefficient of drag
C_L	Coefficient of lift
C_X	Coefficient of axial force
C_Y	Coefficient of side force
C_Z	Coefficient of normal force
$e_i = (x_i, y_i, z_i)$	Inertial, Earth-fixed frame
$e_b = (x_b, y_b, z_b)$	Body-fixed frame
$e_w = (x_w, y_w, z_w)$	Wind-axes frame
Fes	Force elevator stick
Frp	Force rudder pedal
F_{Xaero}	Total aerodynamic force in body-fixed x-axis
F_{Yaero}	Total aerodynamic force in body-fixed y-axis
F_{Zaero}	Total aerodynamic force in body-fixed z-axis
g	Acceleration due to gravity (m/s2)
h	Altitude (feet or m)
Ix	x body moment of inertia (kg·m2)
Ixy	x-y body axis product of inertia (kg·m2)
Ixz	x-z body axis product of inertia (kg·m2)
Iy	y body axis moment of inertia (kg·m2)
Iyz	y-z body axis product of inertia (kg·m2)
Iz	z-body moment of inertia (kg·m2)
m	Aircraft total mass (kg)
M	Mach number
nx, ny, nz	Load factor along x-, y-and z-axes respectively (g)
pb	Body-fixed roll rate (deg/s)

p_{dem}	Demanded roll rate (deg/s)
q_b	Body-fixed pitch rate (deg/s)
q_{dem}	Demanded pitch rate (deg/s)
p_N	position north (m)
p_E	position east (m)
rb	Body-fixed yaw rate (deg/s)
S_{ref}	Wing surface (m2)
t	Time (s)
Tss	Throttle stick setting
u_b, v_b, w_b	Body-fixed velocities along x-, y-and z-axes respectively
$up, uq, u\beta$	Control channel roll, pitch and yaw
V_T	Total velocity (m/s)
x, y, z	Earth axes positions (m)
x_{cg}, y_{cg}, z_{cg}	Center of gravity location along x-, y-and z-axes respectively
x_v, y_v, z_v	Positions in vehicle carried reference frame (m)
α	Angle of attack (deg)
β	Angle of sideslip (deg)
γ	Flight path angle (deg)
δ	Vector of system parameters
δ_{lc}	Left canard deflection (deg)
δ_{ldg}	Landing gear deflection
δ_{le}	Leading edge flap deflection (deg)
δ_{lie}	Left inner elevon deflection (deg)
δ_{loe}	Left outer elevon deflection (deg)
δ_r	Rudder deflection (deg)
δ_{rc}	Right canard deflection (deg)
δ_{rie}	Right inner elevon deflection (deg)
δ_{roe}	Right outer elevon deflection (deg)
φ	Bank angle (deg)
θ	Pitch angle (deg)
ρ	Density of air (kg/m3)
ψ	Heading angle (deg)
(.)	Derivative with respect to time

List of Abbreviations

ADMIRE	Aero-Data Model In Research Environment
AoA	Angle of Attack
ARI	Aileron-Rudder Interconnect
ASA	Adaptive simulated annealing
c.g.	Centre of Gravity
GA	Genetic Algorithm
CGA	Continuous Genetic Algorithm
CS	Control Selector
FCS	Flight Control System
MCS	Multi Coordinate Search
PPM	Pole Placement Method

CLEARANCE OF NONLINEAR FLIGHT CONTROL LAWS USING CONTINOUS GENETIC ALGORITHMS

By
Ali Ghaleb Ali Al-Asasfeh

Supervisor
Dr. Nader Jamil Hamdan, Prof.

Co-Supervisor
Dr. Zaer Salem Abo Hammour

ABSTRACT

Continuous genetic algorithm (CGA) is employed in this paper to solve the nonlinear optimization problem that results from the clearance process of nonlinear flight control laws. This method is used to generate the pilot command signals that govern the aircraft performance around certain points in the flight envelope about which the aircraft dynamics were trimmed. The two pilot command inputs considered in this work are the pitch and roll commands. The performance of the aircraft model due to these two inputs are analyzed to find the worst combination that lead to the exceedance of allowable load factor or angle of attack.

The motivations for using the CGA to solve this type of optimization problem are because the pilot command signals are smooth and correlated, which are difficult to generate using the conventional genetic algorithm (GA). In addition, the CGA has the advantage over the conventional GA method in being able to generate suitable smooth without loss of significant information in the presence of rate limiter in the controller design, and the time delay in response of the actuators. Furthermore, the CGA can be considered as global optimization solver and can deal with multi inputs system.

Simulation results are presented which show superior convergence performance using the CGA compared with conventional genetic algorithms.

Key Words: Clearance of Flight Control Laws, Continuous Genetic Algorithms, ADMIRE.

CHAPTER ONE

INTRODUCTION

1.1 Introduction to clearance process

Modern high performance aircrafts are often designed to be naturally unstable due to performance reasons and, therefore, can only be flown by means of a flight control system, which provides artificial stability. As the safety of the aircraft is dependent on the controller, it must be proven to the clearance authorities that the controller functions correctly throughout the specified flight envelope in all normal and various failure conditions, and in the presence of all possible parameter variations [8].

This task becomes increasingly expensive and time consuming, particularly for high performance aircraft, where many different combinations of flight parameters (e.g. large variations in mass, inertia, centre of gravity positions, highly non-linear aerodynamics, aerodynamic tolerances, air data system tolerances, structural modes, failure cases, etc.) must be investigated so that guarantees about worst-case stability and performance can be made.[16]

The goal of clearance process is to show that a set of selected criteria expressing the stability and handling requirement is fulfilled. The clearance criteria can be divided into categories [8]: (a) linear stability criteria, (b) aircraft handling/Pilot Induced Oscillation (PIO) criteria, (c) non-linear stability criteria, and (d) nonlinear handling criteria. In the clearance process, for each point of the flight envelope, for all possible configurations and

for all combinations of parameter variations and uncertainties, violations of the clearance criteria and the worst-case result for each criterion must be found.

In order to comply with the established clearance criteria the flight control laws are redesigned based on the clearance process results. The current flight clearance process relies on a gridding approach [10, 19], whereby the various clearance criteria are evaluated for all combinations of the extreme points of the aircraft's uncertain parameters. This process is then repeated over a gridding of the aircraft's flight envelope. This is computationally very expensive due to exponential growth of grid points with respect to the dimension of the parameter vector. The behavior between the grid points is not included in the clearance process for gridding method, which may results in bad situation [14].

There is a need to improve flight clearance methods and a promising approach is to use optimization-based methods to find the cases in the flight envelope where the combination of parameter values and/or pilot command inputs performs worst. The objective of such an effort is to design both computationally more efficient and reliable methods than the existing gridding methods.

1.2 Aircraft Dynamic Model

The aircraft model used in this research is the ADMIRE (Aero Data Model In a Research Environment), a nonlinear, six degree of freedom simulation model developed by the Swedish Aeronautical Research Institute using aero data obtained from a generic single seated, single engine fighter aircraft with a delta canard configuration. The aircraft model is augmented with a flight control system and includes engine dynamics and actuator models, number of uncertain parameters.[9]

The aircraft dynamics are modeled as a set of twelve first order coupled non-linear differential equations [9]:

$$\dot{x}(t) = f(x(t), u(t), p) \tag{1.1}$$

$$y(t) = h\big(x(t), u(t)\big) \tag{1.2}$$

Where $x(t)$is the state vector with twelve components (velocity, angle of attack, sideslip angle angular rate vector, attitude angles, and position, …), p represent the uncertain parameters, $y(t)$is the output vector, $u(t)$ is the control input vector, whose components are left and right canard deflection angel, left and right inboard/outboard elevator deflection angel, leading edge flap deflection angle, rudder deflection angle, and vertical and horizontal thrust vectoring. [13]

The control input is determined by:

$$u(t) = g(x(t), y_{ref}(t)) \tag{1.3}$$

Where $g(.,.)$ is a standard flight control law, which is provided with the ADMIRE model, and $y_{ref}(t)$ is the reference demand that consists of the pilot inputs such as Pitch stick demand, Roll stick demand, and Rudder Pedal demand. Equations (1.1-1.3) represent the closed loop dynamics of the aircraft with the flight control law in the loop.

The ADMIRE is augmented with a flight control system, FCS, in order to provide stability and sufficient handling qualities within the operational envelope. The FCS contains a longitudinal and a lateral part. The function of the longitudinal controller is pitch rate control q_{com} below Mach number 0.58 and load factor control n_{zcom} above Mach number 0.62. A blending function is used in the region in between, in order to switch between the two different modes. The longitudinal controller also contains a speed control

V_{Tcom} . The lateral controller enables the pilot to perform roll control around the velocity vector of the aircraft $P_{\omega com}$ and to control the sideslip angle β_{com} .

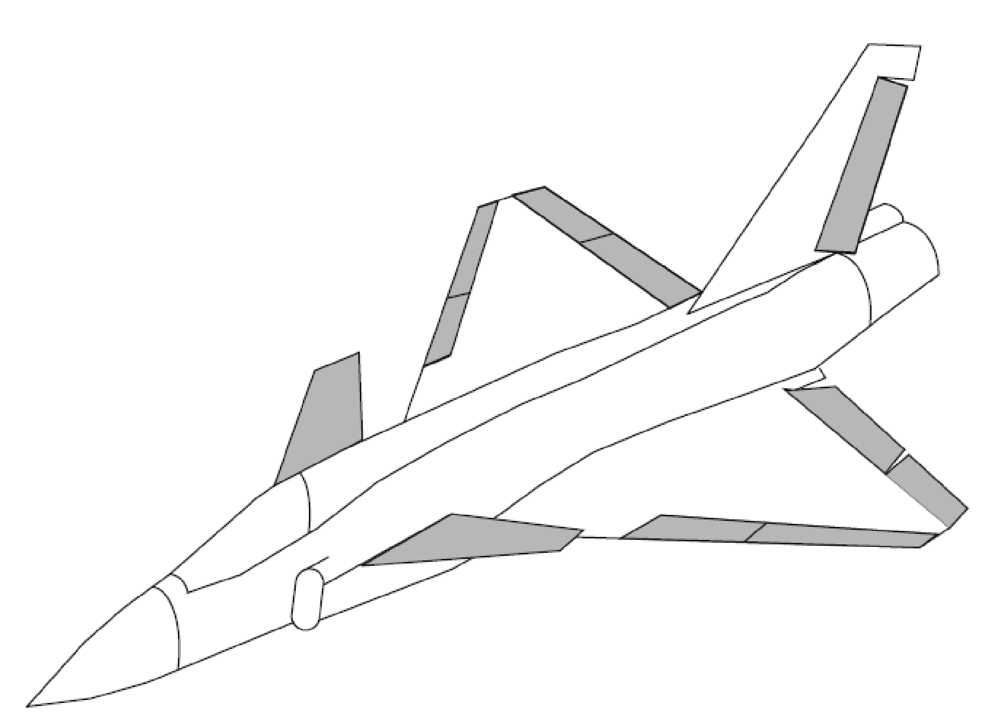


Figure 1.1: Principal layout of the control surface configuration.

1.3 Continuous genetic algorithms (CGAs)

Genetic algorithms, GAs, were developed by John Holland and are modeled on the Darwinian concepts of natural selection and evolution [11]. In genetic algorithms, a population of potential solution is caused to evolve toward a global optimal solution, which occurs because of pressure exerted by the selection process and exploration of the solution space accomplished by crossover and mutation operators [23].

Genetic algorithms are relatively new class of optimization technique, which are generating a growing interest in the engineering community. They are well suited for a broad range of problems encountered in science and engineering. They have performed

efficiently in a number of diverse applications in electrical engineering such as system identification, conventional controller design, fuzzy controller design, optimal control problems, neural networks, and image processing. The principal motivation for researchers and practitioners in genetic algorithms are the following [1, 2]:

1. *Limitation of traditional methods:* The practitioners' motives in genetic algorithms are rooted in the limitation of traditional optimization and operations research methods. A certain optimization methods is well tuned to particular class of optimization problems, but when a problem comes along that violates the assumptions of such methods, solution results can be particularly disappointing.
2. *Methods investment:* The wide spectrum of traditional narrow-band algorithms implies that the practitioner should master a collection of technique rather than a single broadly competent method. For example, for a linear problem with linear constraints, one can use linear programming; for a stage decomposable problem, dynamic programming can be employed. For a nonlinear problem with nonlinear constraints, nonlinear programming can be utilized, and so on. Genetic algorithm, on the other hand, is an optimization procedure that works well over a broader class of optimization problems because the evolution of such natural system takes place via mechanism that are in many ways invariants across species.
3. *Model investment:* Method investment costs can be significant, but for many users the lion share of investment is tied up in modeling or simulation. Most complex optimization involves a sophisticated objective function that may itself rely on models. Prior to using such models for optimization or design, users expend considerable time and effort inputting data, running test cases, tuning the models to agree with the real

world, and then using the models for analysis. After such large investment in modeling, no user likes to be told that in order to perform an optimization, the model must be shortened into a form preferred by a particular optimization method; but many optimization methods require exactly this kind of model transformation. Genetic algorithms on the other hand, take their function evaluations as they come, thereby respecting the significant investment that users may have in analysis model using that model without substantial modification or transformation such as linearization. However, because GAs make relatively few assumptions about the solution space, and because the interface between GAs and evaluations involves only passing function evaluation values, a GA solution may require hundreds or thousands of function evaluations.

The construction of a genetic algorithm for the solution of any optimization problem can be separated into five distinct and yet, related tasks. First, the genetic representation of potential problem solutions. Second, a method for creating an initial population of solution. Third, the design of the genetic operators. Fourth, the definitions of the fitness function. Fifth, the setting of the system parameters, including the population size, probabilities with which genetic operators are applied. The five mentioned factors have resulted in the availability of numerous variants of GAs reported in literature [3,5,12,29].

In relation to the genetic representation of potential problem solutions, GAs are mainly classified into two categories: the binary-coded GAs and the real-coded GAs [5]. In a conventional binary-coded genetic algorithm, binary sub-strings corresponding to each design variable are stacked head to tail to yield a binary string that represents a particular design. In contrast, the real-coded genetic algorithm does not make use of the binary

representation, allowing for gene transformation operations to be conducted on the original real-valued representation of the design variables. Real-coded genetic algorithms are usually used to solve continuous optimization problems. While handling optimization problems, real variants of the algorithms offer a number of advantages over binary-coded schemes. First, the efficiency of GA is increased, as there is no need to convert genotype to phenotypes before each function evaluation. Second, less memory is required as efficient floating-point internal computer representations can be used directly. Third, there is no less in precision by discretization to binary or other values. Fourth, there is more freedom to use different genetic operators [1,5].

Existing crossover schemes are mainly divided into two categories: genotype crossover and phenotype crossover. Genotype crossover is performed by swapping partial strings between the two parents. It is divided into three classes, depending on the number of crossing points that include, single-point crossover, multipoint crossover and uniform crossover schemes. The phenotype crossover is the well-known arithmetic crossover and is performed by interpolating the phenotype values of the two parents. Similarly, the mutation process has two main variants that include the genotype and phenotype mutations. For binary-coded GAs, the genotype mutation is the bitwise complement mutation operator [8]. The phenotype mutation has two variants that include static and dynamic mutations. The dynamic mutation is applied by performing random displacements for the selected variables from its original values, while the static mutation is carried out by assigning a completely new random value to the selected variables. Dynamic mutation is particularly useful in fine-tuning the population in later stages of GA evolution when static mutation might cause too great perturbation and may lead to structures with very low fitness.

Real coded GAs whose operators are applied at the variable level are useful when dealing with continuous optimization problems that do not have any requirement on the continuity and/or smoothness of the resulting solution curves, which is generally the case with most optimization problems [30]. However, when dealing with continuous optimization problems that require the continuity and/or smoothness of the solution curves, the performance of such GAs will be poor. To solve this problem, Gutowski [12] introduced a special type of real coded GAs, smooth genetic algorithms, which are well suited for continuous optimization problems with continuous smooth solution curves. The main area of application of his algorithms is the reconstruction of unknown, continuous, and smooth distributions of various physical quantities derived from experimental data. Smooth genetic algorithms proposed by Gutowski depend on the evolution of curves in one-dimensional space. He used a Gaussian function for initializing the population, tangent hyperbolic crossover function and Gaussian mutation function.

Continuous genetic algorithms (**CGAs**) [1,2,3] was shown to be an efficient method for the solution of optimization problems in which the parameters to be optimized are correlated with each other or the smoothness of the solution curve should be achieved. It has been successfully applied in the motion planning of robot manipulators in the field of robotics [1] and in the numerical solution of boundary value problems in the field of applied mathematics [2].

The conditions about the continuous functions that can be used in such algorithm should be clearly stated. In relation to the initialization function, any smooth function that is close enough to the expected solution curve can be used. It is to be noted that the closer the initialization function to the final solution, the faster the convergence speed since the

coarse tuning stage of CGA in this case will be bypassed and the algorithm will jump to the fine tuning stage. In the case that there is no prior information about the expected solution curves, then any smooth function can be used and a mixture of functions will be beneficial in this case to result in a diverse initial population. The effect of the initial population usually dies after few tens of generations and the selection mechanism, crossover and mutation operators govern the convergence speed after that.

To summarize the evolution process in CGA, an individual is a candidate solution of the required solution curves; that is, each individual consists of N_c solution curves each consisting of N_t variables. This results in a two-dimensional array of the size $N_c \times N_t$. The population undergoes the selection process, which results in a mating pool among which pairs of individuals are crossed with probability P_{ci}. Within that pair of parents, individual solution curves are crossed over with probability P_{cc}. This process results in an offspring generation where every individual child undergoes mutation with probability P_{mi}. Within that child, individual solution curves are mutated with probability P_{mc}. After that, the next generation is produced according to the replacement strategy applied.

This process is repeated until the convergence criterion is met where the N_c solution curves of the best individual are the required solution curves. The final goal of discovering the required solution curves is translated into finding the fittest individual in genetic terms.

It is noted that the GA uses pit representation of system parameters and thus has a problem of convergence if the grid is not relatively small. On the other hand, the CGA, as explained above, uses continuous representation of the system parameters that insures convergence even for large multi input systems.

1.4 Literature review on solving the optimization based-clearance process

The clearance process can be classified in many categories depending on the requirements and handling criteria needed to be clarified.

Several researchers use the fixed pilot command signal to validate the parametric uncertainties set. Prathyush P. Menon *et al* [16,17] describe a hybrid optimization approach to the clearance of flight control laws for highly augmented aircraft. The approach is applied to the problem of evaluating a nonlinear clearance criterion for a detailed simulation model of a high performance aircraft with a delta canard configuration and a full authority flight control law.

Forssel and Hyden [10], validate the functionality of Maneuver Load Limiter in the longitudinal control system for the generic aircraft simulation model ADMIRE, using global optimization algorithms based on Genetic Algorithm and Adaptive Simulated Annealing.

To date, very little research has been reported in the literature on the FCL clearance for time varying pilot command inputs.

D. Skoogh *et al* [24] present approaches for clearance of flight control laws for time varying pilot command inputs, using optimization methods. These approaches are based on the simulation of nonlinear aircraft models, parameterization of pilot command inputs signals and solving the resulting nonlinear optimization problem. The purpose is to find the pilot command signals that result in the worst possible behavior of the aircraft. The studied pilot signals are lower order piecewise polynomials, and the resulting parameter set is a real valued finite dimensional set on which global and local optimization methods are studied.

Ryan G. W. III [21] evaluates the departure susceptibility of the X-31 Enhanced Fighter Maneuverability demonstrator aircraft. Genetic Algorithms (GA's) and a high-fidelity nonlinear simulation model were used to search for pilot command inputs that maximized a cost function associated with aircraft departures – the absolute sum of certain states of the system, such as attitude rates, angle of attack (AOA) and sideslip angle.

This genetic algorithm approach was subsequently developed using a multi-modal genetic search approach with a refined energy-like cost function, and applied to a full nonlinear simulation model of the Indian Light Combat Aircraft, [4]. Neither of the above studies, however, considered any form of uncertainty in the aircraft simulation model.

A particular sequence of pilot command inputs called the Clonk maneuver, which was developed by Saab using piloted simulation testing to detect the departure susceptibility of the Gripen aircraft, was applied to the ADMIRE aircraft simulation model, [15]. Global optimization methods were then used to compute the worst-case combination of uncertain aircraft parameters for this sequence of command inputs.

The resulting nonlinear optimization problem by using the above mentioned signals were solved using genetic algorithm and differential evolution algorithm. These two approaches apply many restrictions on the shape of the pilot command inputs. Due to the presence of rate limiter in the controller design, and the time delay in response of the actuators it is suitable to use smooth and correlated. Other investigators treated the pilot command inputs as predetermined step inputs and evaluated the clearance criteria based on evaluation of system parameter uncertainties.

Also, all previously reported work on optimization-based flight clearance has focused on the clearance of flight control laws at single points, rather than over continuous regions of the aircraft flight envelope, [8,15,21,22].

1.5 Thesis objective

Thesis objective summarized in the following: Continuous genetic algorithms will be used as an optimal control problem solver to validate the flight control laws for ADMIRE benchmark aircraft model. Clearance process will focus on the nonlinear aircraft handling criteria; those criteria which results in the worst possible behavior of the aircraft (normal load factor, n_z or angle of attack, α exceedance), figure (1.2) shows the block diagram of the clearance process. The flight control laws consist of the Lateral and Longitudinal controllers; both controllers will be subject to clearance process.

The expected advantage concerns mainly the ability to find the global optimal solution for nonlinear optimal problem having multiple solutions such as the clearance process found in flight control laws validation.

To the author's knowledge application of this technique to flight control laws clearance process has been not addressed in the open literature.

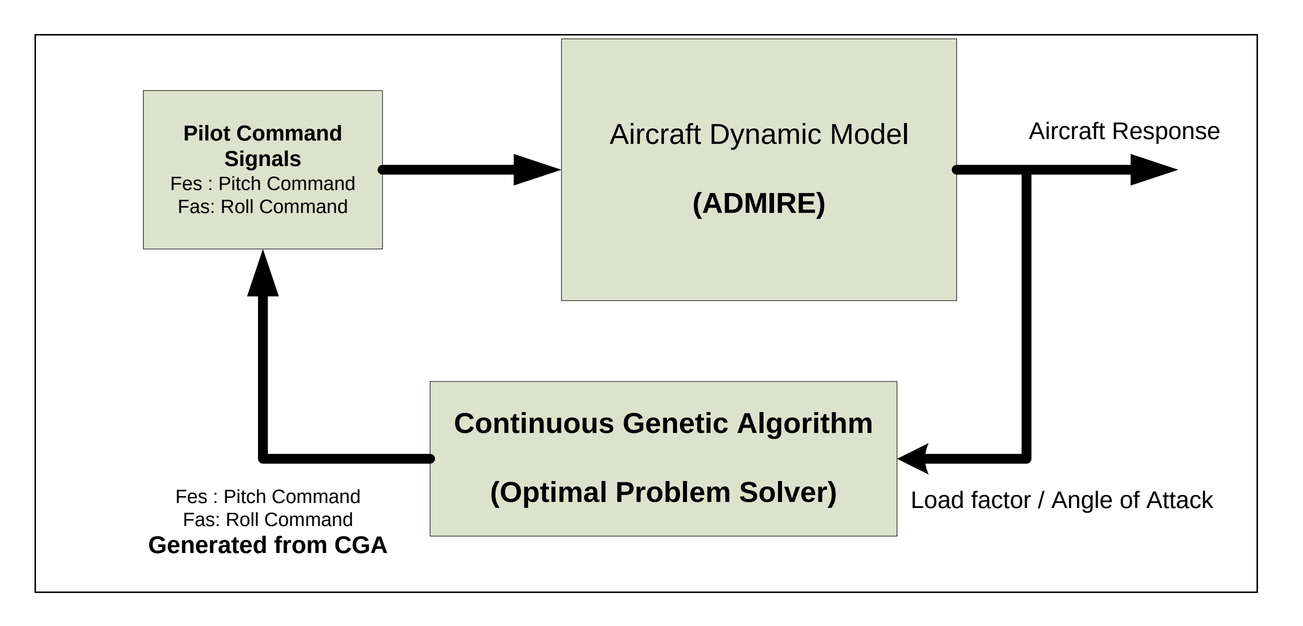


Figure 1.2: Block diagram of the clearance process

1.6 Organization of the thesis

The thesis is organized in several chapters. Chapter two presents the nonlinear flight control laws clearance and discusses the related topics related to the clearance process including the relevant work reported in the literature regarding the flight control laws clearance. Chapter three present the aircraft dynamic model (ADMIRE) which was used in this thesis. Chapter four presents the Continuous Genetic Algorithms. Chapter five presents the results of the simulation for flight control laws clearance using the Continuous Genetic Algorithm. Finally, the conclusions and some suggestions for future research are presented in chapter six.

CHAPTER TWO

NONLINEAR FLIGHT CONTROL LAWS CLEARANCE

2.1 Design process of flight control system

Flight control systems (FCS) are very important to govern the aircraft response. The reliability and the accuracy of FCS guarantee to guide the aircraft to the required route with economic operation. The FCS coordinates the deflection of the flight control surfaces to get the desired movement depending on the pilot inputs.

Old aircraft has no flight control surfaces as it now, the pilot command were transferred directly to flight control surfaces via mechanical connections without intermediate system. Due to the industry improvement new equipment were founded such as the hydraulic actuator, synchronic devices, and advanced controllers that can be added to the aircraft system to reduce the required forces from the pilot and to automate these system in order to improve the response and the accuracy of the system overall. The latest technology in flight control system is Fly by Wire, this technology eliminate the mechanical connection and use electrical signals instead of it.

All of these improvements on the FCS and the demand for high performance aircraft make the FCS more and more complex due to the presence of electronics system and the redundancy of the flight control surfaces and controllers. However, the FCS design becomes complex and the design need to be validated in order to put the aircraft in service.

The FCL design process is divided to the following [22]: the first phase is Off-line design, the flight control laws that govern the aircraft performance are introduced at this

stage. This stage includes also the parameters tuning process to get the required response to comply with the required specifications. The second phase is Pilot-in-the-loop simulation phase, in which we perform the handling qualities assessment. The third phase is iron-bird phase, in which we make sure that all the FCL components are operate as required. the fourth phase is the core of this research which is the clearance phase. In this phase we approve that the designed controller will operate in safe and meat the authority requirements for all possible failure condition and for all parameters variations. Finally test flight phase performed to validate the aircraft systems.

2.2 The importance of clearance process

Aviation industry use the highest technology during design of flight controllers. This technology enable the design engineers to study and analysis the flight controller in order to define any challenges and failure that can affect the performance of the controller [25]. However, the development of flight control laws is very complex, multi-disciplinary task and the many problems that have to be solved make it a costly and lengthy process [10].

Researchers make use of existing mathematical methods and develop new methods for flight control laws development process specially for the clearance process which found at the last stage of the flight control laws design process. In the clearance process it should be proven to the authorities that the flight controller design meet assigned specifications and can guide the aircraft to fly safely throughout the flight envelope and for all failure conditions and for any parameter variations.

In clearance process many different configurations have to be studied including mass, inertia, and center of gravity location variation. In addition, the error tolerances in the aerodynamic data and on air data signals used for control law scheduling, have to be taken into account [10].

The clearance should investigate each point of the flight envelope, for all possible configurations and for all combinations of parameter variations and uncertainties, violations of the clearance criteria and the worst-case result for each criterion have to be founded. Based on the clearance results, the safe operation area in the flight envelope is defined and the required modification the flight controller can set. In some cases flight operation restriction can be set to avoid the unsafe region in the flight envelope, these restriction might involve switching to the backup system.

Researchers have been develop several mathematical technique for clearance process analysis. Each of these techniques has its known strengths and weaknesses. However, at this moment it is still difficult for the aeronautical industry to assess whether their application would improve the efficiency of the FCL clearance process or not. [8]

2.4 Optimization-based flight clearance process

The clearance process can be handled from deferent points of view such as the stability analysis based on the existed control theory can be used to clear the flight control system or optimization based clearance method. In optimization based clearance method the clearance process is reformulated to optimization problem to find the combination of pilot command inputs or parameters that cause large variation in the stability/ performance criteria [14,17].

The optimization problem use the analyzed criteria to define performance index (distance function, (FC, P) , between the accepted criteria values and the non accepted values) to find worst-case induced by the pilot input combinations or parameters in an optimization-based clearance approach.[8]

The clearance task for the AoA/n_z- limit exceedance criterion can be formulated as the question: which combination of pilot command input, given a specific flight condition, will result in the highest α and n_z of the ADMIRE? This question can then be regarded as an anti-optimization problem if α and n_z are considered as distance functions that are allowed to vary over a parameter space. Then the optimization algorithm, that tries to maximize α or/and n_z, will search over the allowed space and find the combination of parameters that results in the highest α or/and n_z, which is then the worst case of the AoA/n_z - limit exceedance criterion.

For the AoA/n_z-limit exceedance criterion, two different distance functions can be defined [8]:

$$\alpha_{max}\,(FC, P) = \max_{t \in T} \alpha(FC, P, t) \tag{2.1}$$

$$n_{z_{max}}\,(FC, P) = \max_{t \in T} n_z(FC, P, t) \tag{2.2}$$

where the functions α and n_z are the output signals derived from the ADMIRE model, for the flight condition FC, which consist of the aircraft speed and the altitude in which the aircraft model will be trimmed, with the model input parameter vector P defined as:

$$P = \left(P_{input}\,, \Delta\right) \tag{2.3}$$

Where, P_{input} is the command input from the pilot which can be one or combination of the Pitch, Roll, and Yaw command signal (Pitch stick force, Roll stick force, and Rudder pedal input). Moreover, Δ is the model uncertainty parameter vector.

For a given flight, condition (*FC*) an optimization problem can then be formulated:

$$p_{\alpha}^{*} = \max_{p\in P} \alpha_{max}(p) \quad (2.4)$$

$$p_{n_z}^{*} = \max_{p\in P} n_{z_{max}}(p) \quad (2.5)$$

Where the solutions p_{α}^{*} and $p_{n_z}^{*}$ are the pilot command input.

2.5 Solving flight clearance optimal problem

The flight clearance problem that formulated, as maximization problem need to be solved using the proper and most efficient optimization algorithm. The optimization problem itself is to find the pilot command inputs that gives the worst violation of the criterion defined in equation (2.4, 2.5).

Since the clearance process should handle all the possible aircraft configuration over all the point in the flight envelope it is required to use the most computationally efficient approach to solve the optimization problem. Many different classes of optimization algorithms are available in the literature. Some of these algorithms use the gradient information of the cost function to find the search direction while determining the optimum. Other algorithms use only the cost function value. Many of these non-gradient-based search and optimization techniques make use of heuristic search directions, in an efficient and intelligent way [16].

Local optimization methods such as SQP (Sequential Quadratic Programming), and L-BFGS-B (Limited memory Broyden-Fletcher-Goldfarb-Shanno method with bounded constraints) [5] were used to evaluate a range of linear clearance criteria for the HIRM+ (High Incidence Research Model) aircraft model.

Global optimization schemes such as genetic algorithms (GAs), adaptive simulated annealing (ASA) and multi coordinate search (MCS) were also applied to evaluate nonlinear clearance criteria for both ADMIRE model and HIRM+ model.

2.6 The Clearance Requirements and Criteria

The clearance requirement based on the ADMIRE aircraft model can be summarized in the following based on the data presented in [14]. The aircraft shall be highly maneuverable with a maximum roll performance of the order of 300◦/s at high subsonic speed and 1 g, a maximum normal load factor of 9 g, and a maximum angle of attack of 30◦. There may be difficulties with lateral stability at high angle of attack, which could result in a high angle of sideslip. If that is encountered, the maximum angle of attack can be reduced to 26◦. The minimum normal load factor shall be -3 g and the minimum angle of attack -10◦. The FCS shall limit the angle of attack and the normal load factor. The turn performance shall be the highest possible (maximum pitch rate at least in the order of 25◦/s) at corner speed. Corner speed is defined as the speed where the available instantaneous turn rate is the highest, but also the speed where both maximum angle of attack and maximum normal load factor can be achieved as illustrated in Figure 2.1. The corner speed, in terms of Mach number, will vary with altitude. Above corner speed, maximum normal load factor will limit the pitch performance and below corner speed, maximum angle of attack will limit the pitch performance.

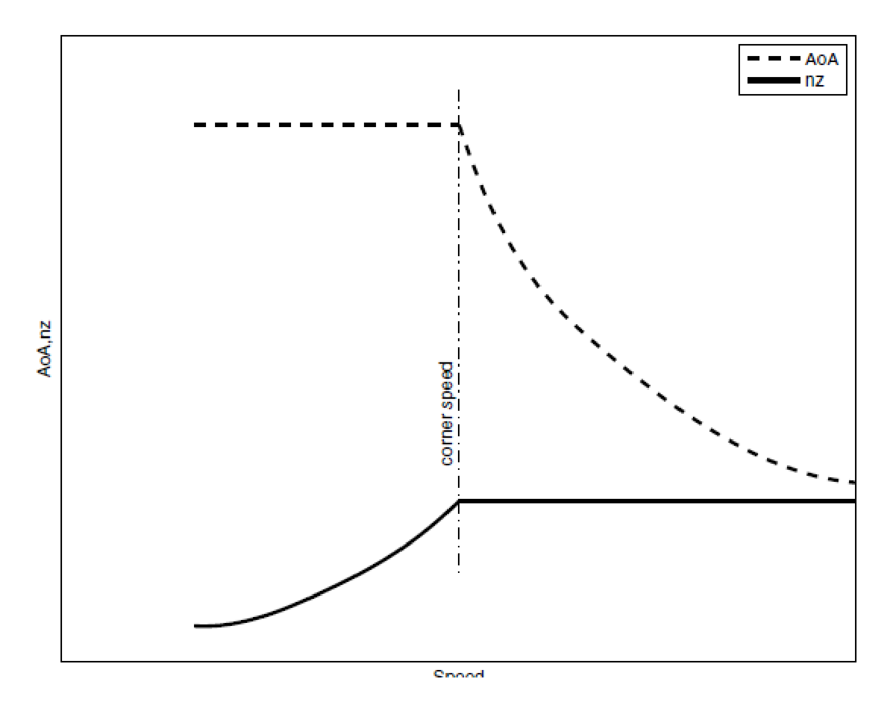


Figure: 2.1: Angle of attack and normal load factor limits versus speed (corner speed is marked with a line) [14].

The design envelope is from Mach 0.3 up to 1.4 and altitude from 100 m up to 6000 m, approximate flight envelope for the design shown in figure 2.2. The design shall focus on the altitude 1000 m and 3000 m. Transonic (Mach number in the range 0.9 to 1.1) is an important part of the flight envelope for the Flight Control Laws. Any pitch-up transients due to speed changes shall be minimized. This is of most interest in the transonic region and around corner speed. The design shall cover an angle of attack from -10° to $+30^{\circ}$, normal load factor from -3 g to +9 g and angle of sideslip from -10° to $+30^{\circ}$. The angle of sideslip shall be kept small during roll maneuvers. The pitch stick input shall be from -40N to +80N and the roll stick input shall be from -80N to +80N.

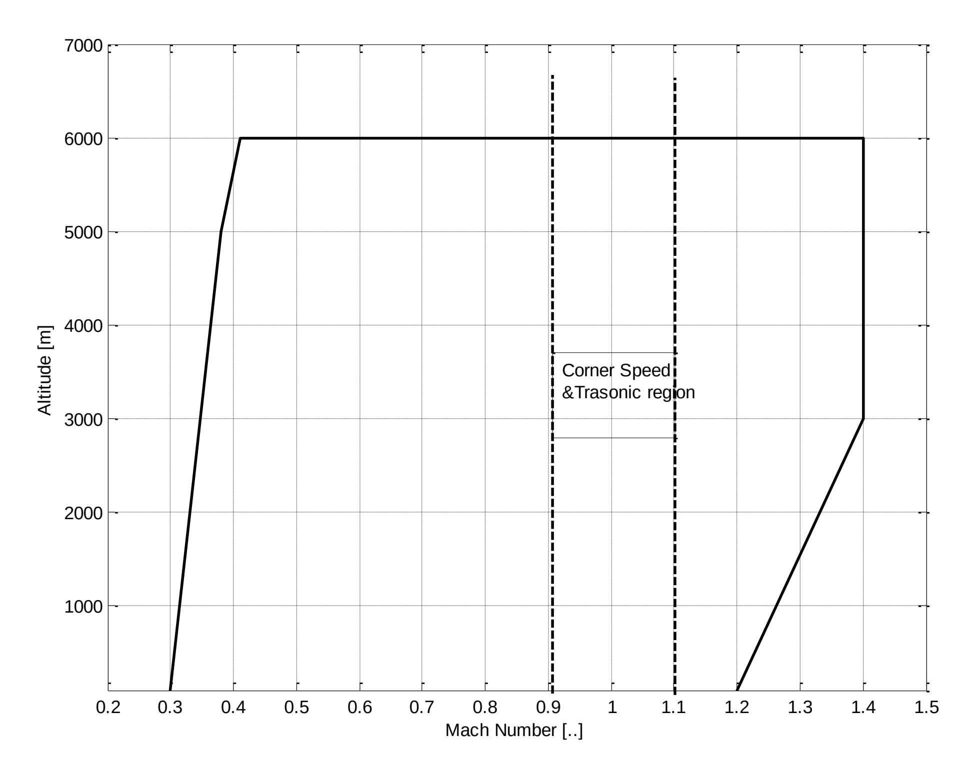


Figure: 2.2: Approximate flight envelope for the design (Corner speed and transonic region are illustrated).

2.7 Chapter Summary

In this chapter the importance of research on the flight control laws clearance were discussed. This shows that it is important to clear the flight control system and approve that it will function in stable and safe operation. All of that is necessary to get the approval from the authorities.

In next section, the design and validation of the flight control laws were discussed. The design steps were shown and the importance of clearance process during design steps to adjust the design and improve it to meet the stability and handling requirement.

The optimization based clearance process and the proper and most efficient optimization algorithm were shown. Finally, the clearance requirement based on the ADMIRE aircraft model were summarized.

CHAPTER THREE

AIRCRAFT DYNAMIC MODEL (ADMIRE)

3.1 Introduction to Aerodynamics

In order to understand aircraft control and behavior, a brief introduction to aerodynamics is essential. Any aircraft motion is determined by the moments and aerodynamic forces acting on the aircraft presented in the proper coordinate and accepted reference. There are two main frame systems which describe the aerodynamic of aircraft; body-fixed frame and earth-fixed frame as shown in figure (3.1) in addition to wind-axis frame. The mathematical model of the aircraft flight dynamic is commonly formulated using these frames.

The mathematical model of flight dynamics describes both a physical system and a process. The equations of the model define relationships between system variables. The model includes mathematical descriptions of flight dynamics, the coordinate frames against which values of system variables take meaning, and the manner in which control over aircraft motion is affected. In what follows a brief description of the mathematical model formulation is presented and closely follows description presented in [9,13].

In the earth-fixed frame, the three axes are pointing north, east and down. This frame is useful for describing the position and orientation of the aircraft. In the body-fixed frame the 3 axes with origin point at the aircraft centre of gravity are pointing forward, over the right wing and down. In this frame, the inertia matrix of the aircraft is fixed thus making the frame suitable for describing angular motions.

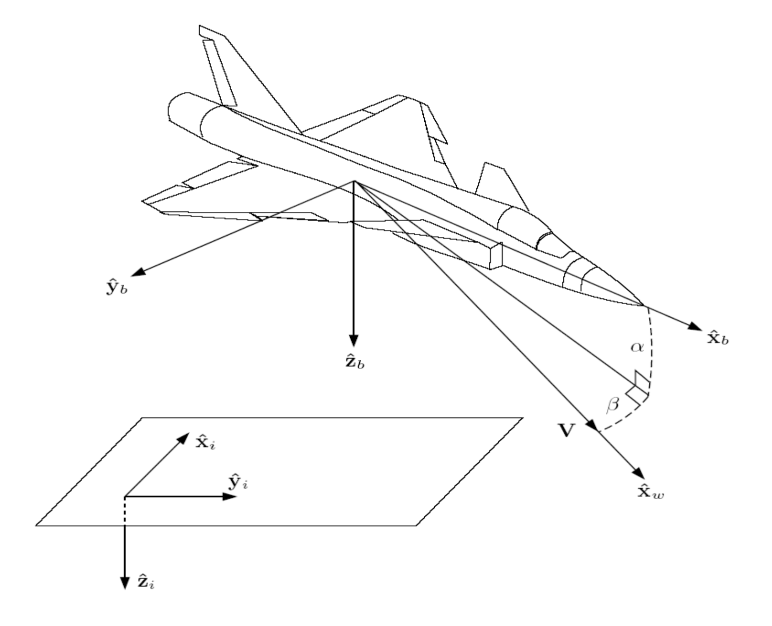


Figure 3.1: Earth fixed frame (i), and body fixed frame (b)

The wind-axes frame (w) derives its x-axis from the velocity vector of the aircraft (V). The wind-axis frame is relative to the fixed-body frame by the angle of attack (α) and the angle of sideslip (β) as shown in Figure 3.1.

$$v = e_b v_b = e_w v_w \tag{3.1}$$

its component vectors in the first two frames are related by:

$$v_w = T_{wb} v_b \tag{3.2}$$

$$v_b = T_{bw} v_w = T_{wb}^T v_w \tag{3.3}$$

Where:

$$T_{wb} = \begin{bmatrix} \cos\beta & \sin\beta & 0 \\ -\sin\beta & \cos\beta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\alpha & 0 & \sin\alpha \\ 0 & 1 & 0 \\ -\sin\alpha & 0 & \cos\alpha \end{bmatrix} = \begin{bmatrix} \cos\alpha\cos\beta & \sin\beta & \sin\alpha\cos\beta \\ -\cos\alpha\sin\beta & \cos\beta & -\sin\alpha\sin\beta \\ -\sin\alpha & 0 & \cos\alpha \end{bmatrix}$$

Considering the aircraft as a rigid body, its motion can be described by its position, orientation, velocity and angular velocity over time. The position vector is given by:

$$p = e_b (p_N \quad p_E \quad -h)^T \tag{3.4}$$

In the earth-fixed frame where p_N = position north, p_E = position east and h = altitude. The orientation of the aircraft can be represented by the Euler angles:

$$\Phi = (\phi \quad \theta \quad \psi)^T \tag{3.5}$$

where ϕ = roll angle, θ = pitch angle and ψ = yaw angle. These angles relate the body-fixed frame to the earth-fixed frame as shown in figure (3.2).

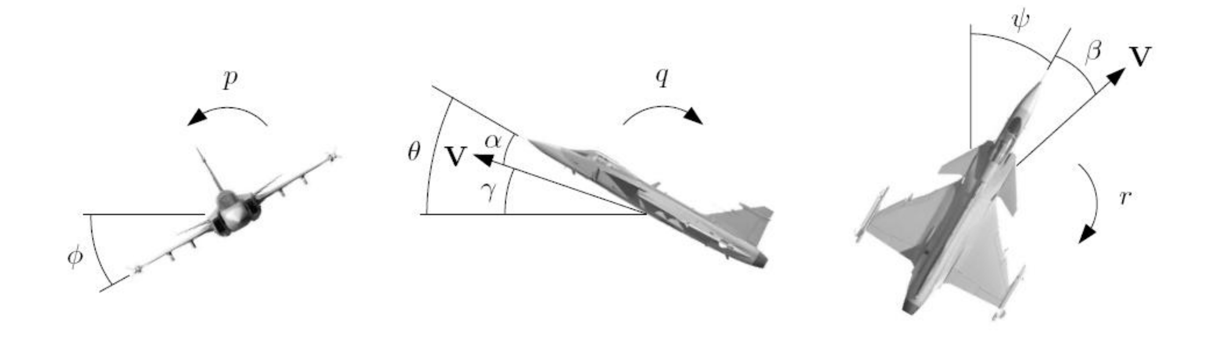


Figure 3.2: Illustration of aircraft orientation angles (φ, θ, ψ) and angular rates (p,q,r)

The velocity vector (V) is given by:

$$V = e_b V = e_w V_w \tag{3.6}$$

$$V = (u \quad v \quad w)^T \tag{3.7}$$

$$V_w = (V_T \quad 0 \quad 0)^T \tag{3.8}$$

in the body-fixed and in the wind-axes coordinate frames respectively. Here u = longitudinal velocity, v = lateral velocity, w = normal velocity and V_T = total velocity (airspeed).

$$V = T_{bw} V_w = V_T (\cos\alpha\cos\beta \quad \sin\beta \quad \sin\alpha\cos\beta)^T \tag{3.9}$$

Conversely, we have that

$$V_T = \sqrt{u^2 + v^2 + w^2} \tag{3.10}$$

$$\alpha = \arctan\frac{w}{u} \tag{3.11}$$

$$\beta = \arctan\frac{v}{V_T} \tag{3.12}$$

when β = ϕ = 0 the flight path angle is defined by:

$$\gamma = \theta - \alpha \tag{3.13}$$

The angular velocity for vector ω is given by:

$$\omega = e_b\omega = e_w\omega_w \tag{3.14}$$

$$\omega = (p \quad q \quad r)^T \tag{3.15}$$

$$\omega_w = T_{wb}\,\omega = (p_w \quad q_w \quad r_w)^T \tag{3.16}$$

In the body-fixed and wind-axes coordinates respectively. p = roll rate, q = pitch rate and r = yaw rate. The wind-axes roll rate p_w is also known as the velocity vector roll rate since it is parallel to the velocity vector **V**

The control variables of an aircraft consist of the thrust produced from the engine combined with the control surfaces of the aircraft such as rudder, aileron and elevator. The deflection of the control surfaces produces aerodynamic forces when airflow is forced across them. The engine produces the speed control while the movement in pitch, yaw and roll is determined by the deflections in the control surfaces (δ) [20].

For both redundancy and performance, concerns modern aircraft typically implement more than three control surfaces see figure (3.3). Roll control is achieved by deflecting the elevons differentially. Pitch control is achieved by combining symmetric elevon deflection that generates a non-minimum phase response with deflection of the canards that produces a response in the commanded direction immediately.

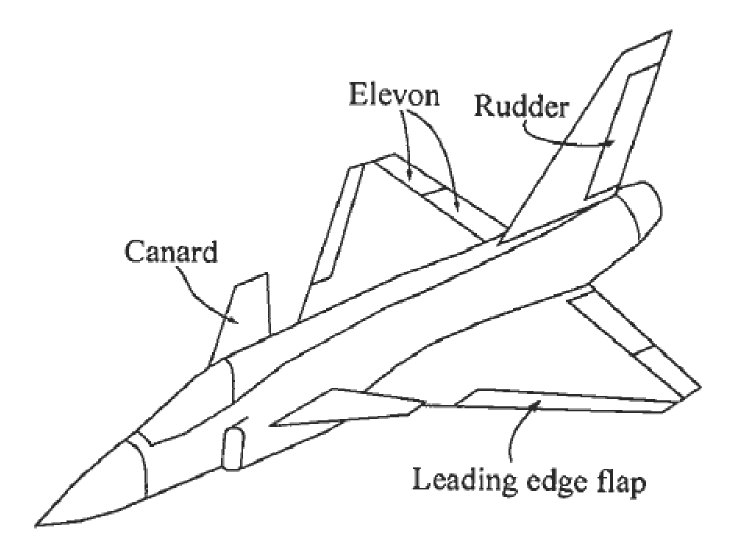


Figure 3.3: Modern delta canard fighter aircraft

The interest in higher angles of attack has founded the development of thrust vectoring. By mounting deflectable vanes at the engine exhaust it is possible to direct the exhaust to provide additional pitching or yawing moments.

Now we can derive a model of the aircraft dynamics. By considering the aircraft as a rigid body allows us to use Newton's laws of motion to investigate the effects of the external forces and moments acting on the aircraft. In the earth-fixed frame (i), Newton's second law states that:

$$\boldsymbol{F} = \frac{d}{dt}\bigg|_i (m\boldsymbol{V}) \tag{3.17}$$

$$\boldsymbol{T} = \frac{d}{dt}\bigg|_i \boldsymbol{H} \tag{3.18}$$

Where $\boldsymbol{F}$ = total force, $\boldsymbol{T}$ = total torque, m = aircraft mass and $\boldsymbol{H}$ = angular momentum of the aircraft.

Using Figure (3.1) allows us to perform the differentiation in the body-fixed frame instead.

$$\boldsymbol{F} = \frac{d}{dt}\bigg|_i (m\boldsymbol{V}) + \omega \times m\boldsymbol{V} \quad (3.19)$$

$$\boldsymbol{T} = \frac{d}{dt}\bigg|_i \boldsymbol{H} + \omega \times \boldsymbol{H} \quad (3.20)$$

As this frame is relative to the aircraft, the inertia matrix I is constant. The angular momentum can be expressed as:

$$\boldsymbol{H} = \mathbf{e_b} I w \quad (3.21)$$

$$\text{Where} \quad I = \begin{bmatrix} I_x & 0 & -I_{xz} \\ 0 & I_y & 0 \\ -I_{xz} & 0 & I_z \end{bmatrix} \quad (3.22)$$

The zero-entries are a property of the aircraft symmetry around the xz-axis. Expressing all vectors in the body-fixed frame gives the following standard equations for rigid body motion in terms of velocity and angular velocity:

$$F = m\big(\dot{V} + \omega \times V\big)g \quad (3.23)$$

$$T = I\dot{\omega} + \omega \times I\omega \quad (3.24)$$

Pitch, yaw and roll angle dynamics during level flight are given by:

$$\dot{\phi} = p \quad (3.25)$$

$$\dot{\theta} = q \quad (3.26)$$

$$\psi = r \quad (3.27)$$

In equations (3.17, 3.18), F and T represent the sum of forces and moments acting on the aircraft at the centre of gravity. These forces are a combination of three major forces; gravity, engine thrust and aerodynamic effects. F and T can therefore be expressed as:

$$F = F_G + F_E + F_A \quad (3.28)$$

$$T = T_E + T_A \quad (3.29)$$

We will now briefly investigate these components. Gravity only gives a force contribution since it acts at the aircraft center of gravity. The gravitational force *mg* is directed along the normal of the earth plane and is considered independent of the altitude. This gives:

$$\mathbf{F}_{\mathrm{G}} = \mathbf{e}_i \begin{pmatrix} 0 \\ 0 \\ mg \end{pmatrix} = \mathbf{e}_b mg \begin{pmatrix} -sin\theta \\ sin\phi cos\theta \\ cos\phi sin\theta \end{pmatrix} = \mathbf{e}_w m \begin{pmatrix} g_1 \\ g_2 \\ g_3 \end{pmatrix} \tag{3.30}$$

where:

$$g_1 = g(-cos\alpha \cdot cos\beta \cdot sin\theta + sin\beta \cdot sin\phi \cdot cos\theta + sin\alpha \cdot cos\beta \cdot cos\phi \cdot cos\theta$$

$$g_2 = g(cos\alpha \cdot sin\beta \cdot sin\theta + cos\beta \cdot sin\phi \cdot cos\theta - sin\alpha \cdot sin\beta \cdot cos\phi \cdot cos\theta)$$

$$g_3 = g(sin\alpha \cdot sin\theta + cos\alpha \cdot cos\phi \cdot cos\theta)$$

using rotation around the 3 axes in Figure (3.2).

The thrust force produced by the engine is denoted by $\mathrm{F_T}$. Assuming the engine is positioned to produce a force parallel to the aircraft body axis gives:

$$\mathbf{F}_{\mathrm{E}} = \mathbf{e}_b \begin{pmatrix} \mathrm{F_T} \\ 0 \\ 0 \end{pmatrix} \tag{3.31}$$

Also assuming the engine is positioned so the thrust point lies in the xz-plane of the body-fixed frame offset from the center of gravity by $\mathrm{z_{TP}}$ along the z-axis gives a pitching moment:

$$\mathbf{T}_{\mathrm{E}} = \mathbf{e}_b \begin{pmatrix} 0 \\ \mathrm{F_T z_{TP}} \\ 0 \end{pmatrix} \tag{3.32}$$

The aerodynamic forces and moments are generated by the interaction between the aircraft body and the surrounding air. The size and direction of the moments are determined by the amount of air diverted by the aircraft in different directions. The amount of air directed by the aircraft is determined by:

• The speed and density of the airflow (V_T, ρ)

• The geometry of the aircraft: S (wing area), b (wingspan), c (mean aerodynamic chord)

• The orientation of the aircraft relative to the airflow: α, β

• The control surface deflections: δ

The aerodynamic forces and moments also depend on other variables, such as angular rates (p, q, r) and the time derivatives of the aerodynamic angles ($\dot{\alpha}, \dot{\beta}$) but these effects are not as prominent. This motivates a standard way of modeling scalar aerodynamic forces and moments:

$$Force = \overline{q}SC_F(\delta, \alpha, \beta, p, q, r, \dot{\alpha}, \dot{\beta}) \tag{3.33}$$

$$Moment = \overline{q}SlC_M(\delta, \alpha, \beta, p, q, r, \dot{\alpha}, \dot{\beta}) \tag{3.34}$$

3.2 Description of ADMIRE

The ADMIRE is a generic model of a small single-seat fighter aircraft with a delta-canard configuration. The ADMIRE contains twelve states (V_T , α, β, p_b,q_b, r_b, φ, θ, ψ, x_v, y_v, z_v) plus additional states due to the presence of actuators, sensors and Flight Control System (FCS). Available control surfaces are left and right canard, leading edge flaps, four elevons, rudder, and throttle setting. The model is also equipped with thrust vectoring capability, air brakes and a choice to have the landing gear up or down [13]. In what follows a brief description of ADMIRE is presented, which is for clarity purposes, closely follows description presented in [9,13].

The ADMIRE include an FCS in order to provide stability and sufficient handling qualities within the operational envelope. The FCS contains a

longitudinal and a lateral part. The longitudinal controller provides pitch rate control below Mach number 0.58. For Mach numbers greater than or equal to 0.62 it provides load factor control. The corner speed is close to Mach number 0.60. In the Mach number region between 0.58 and 0.62, a blending is performed. There is an α-limiter functionality active during pitch rate mode. The longitudinal controller also contains a very rudimentary speed controller. The lateral controller enables the pilot to perform initial roll control around the velocity vector of the aircraft and angle of sideslip control. Sensor models used by the FCS are incorporated in the model, together with a 20 ms computer delay on the actuator inputs, implemented as Padé approximations. There is also a possibility to vary some parameters and uncertainties within given tolerances, in order to facilitate the robustness analysis [9].

3.2.1 Model Data and Envelope

Aircraft configuration data used in ADMIRE are described in Table 3.1. The data gives an idea about the size of the aircraft. The mass and inertia are functions of the percentage of internal fuel onboard, and in the nominal case, the mass represents 60% of fuel loaded.

The ADMIRE mass center is located at the aerodynamic reference point on the x-axis and slightly above the z-axis. There is of course a possibility to change the values of the mass parameters, i.e., the mass, mass distribution and the position of the mass center, which affect the stability margin of the aircraft. The FCS is designed for the nominal model described in Table 3.1 [13].

Table 3.1: Nominal configuration data.

Parameter :	Value:	Unit:
wing area	45.00	m^2
wing span	10.00	M
wing chord(mean)	5.20	M
Mass	9100	Kg
Ix	21000	kgm^2
Iy	81000	kgm^2
Iz	101000	kgm^2
Ixz	2500	kgm^2

The ADMIRE flight envelope is restricted to Mach numbers less than 1.4 and altitudes below 6 km. Within the flight envelope, there are additional constraints due to aero-data.

3.2.2 Aircraft Dynamic Model

The aircraft dynamics are modeled as a set of twelve first order non-linear differential equations on the form

$$x = f(x, u, p) \quad (3.35)$$

$$y = g(x, p) \quad (3.36)$$

where x is the state vector, u is the input vector, y is the output vector and p is the uncertainty parameter space vector. The state equations used are listed below:

$$\dot{V_T} = (u_b \cdot \dot{u_b} + v_b \cdot \dot{v_b} + w_b \cdot \dot{w_b})/V_T \quad (3.37)$$

$$\dot{\alpha} = (u_b \cdot \dot{w}_b - w_b \cdot \dot{u_b})/(u_b^2 + w_b^2) \quad (3.38)$$

$$\dot{\beta} = (\dot{v_b} \cdot V_T - v_b \cdot \dot{V_T})/(V_T^2 \cdot cos\beta) \tag{3.39}$$

$$\dot{p_b} = (C_1 \cdot r_b + C_2 \cdot p_b) \cdot q_b + C_3 \cdot M_x + C_4 \cdot M_z \tag{3.40}$$

$$\dot{q_b} = C_5 \cdot p_b \cdot r_b - C_6(p_b^2 - r_b^2) + C_7 \cdot M_y \tag{3.41}$$

$$\dot{r_b} = (C_8 \cdot p_b - C_2 \cdot r_b)q_b + C_4 \cdot M_x + C_9 \cdot M_z \tag{3.42}$$

$$\dot{\psi} = (q_b \cdot sin\phi + r_b \cdot cos\phi)/cos\theta \tag{3.43}$$

$$\dot{\theta} = q_b \cdot cos\phi - r_b \cdot sin\phi \tag{3.44}$$

$$\dot{\phi} = p_b + tan\theta \cdot (q_b \cdot sin\phi + r_b \cdot cos\phi) \tag{3.45}$$

$$\dot{x_v} = cos\theta \cdot cos\psi \cdot u_b + (sin\phi \cdot sin\theta \cdot cos\psi - cos\phi \cdot sin\psi) \cdot v_b + (cos\phi \cdot sin\theta \cdot cos\psi + sin\phi \cdot sin\psi) \cdot w_b \tag{3.46}$$

$$\dot{y_v} = cos\theta \cdot sin\psi \cdot u_b + (sin\phi \cdot sin\theta \cdot sin\psi + cos\phi \cdot cos\psi) \cdot v_b + (cos\phi \cdot sin\theta \cdot cos\psi + sin\phi \cdot sin\psi) \cdot w_b \tag{3.47}$$

$$\dot{z_v} = -sin\theta \cdot u_b + sin\phi \cdot cos\theta \cdot v_b + cos\phi \cdot cos\theta \cdot w_b \tag{3.48}$$

where

$$\dot{u_b} = r_b \cdot v_b - q_b \cdot w_b - g_0 \cdot sin\theta + F_x/m \tag{3.49}$$

$$\dot{v_b} = -r_b \cdot u_b + p_b \cdot w_b + g_0 \cdot sin\phi \cdot cos\theta + F_y/m \tag{3.50}$$

$$\dot{w_b} = q_b \cdot u_b - p_b \cdot v_b + g_0 \cdot cos\phi \cdot cos\theta + F_z/m \tag{3.51}$$

$$C_1 = \left((I_y - I_z) I_Z - I_{xz} I_{xz} \right) / \Gamma \tag{3.52}$$

$$C_2 = \left((I_y - I_y + I_Z) I_{xz} \right) / \Gamma \tag{3.53}$$

$$C_3 = I_z / \Gamma \tag{3.54}$$

$$C_4 = I_{xz} / \Gamma \tag{3.55}$$

$$C_5 = (I_z - I_x) / I_y \tag{3.56}$$

$$C_6 = I_{xz} / I_y \tag{3.57}$$

$$C_7 = 1 / I_y \tag{3.58}$$

$$C_8 = \left(I_x (I_x - I_y) - I_{xz} I_{xz} \right) / \Gamma \tag{3.59}$$

$$C_9 = I_x / \Gamma \tag{3.60}$$

$$C_{10} = I_x I_z - I_{xz} I_{xz} \tag{3.61}$$

The output vector consists of the state variables plus additional variables defined by the following equations:

$$u_b = V_T \cdot cos\alpha \cdot cos\beta \tag{3.62}$$

$$v_b = V_T \cdot sin\beta \tag{3.63}$$

$$w_b = V_T \cdot sin\alpha \cdot cos\beta \tag{3.64}$$

$$u_v = cos\theta \cdot cos\psi \cdot u_b + (sin\phi \cdot sin\theta \cdot cos\psi - cos\phi \cdot sin\psi) \cdot v_b + (cos\phi \cdot sin\theta \cdot cos\psi + sin\phi \cdot sin\psi) \cdot w_b \tag{3.65}$$

$$v_v = cos\theta \cdot sin\psi \cdot u_b + (sin\phi \cdot sin\theta \cdot sin\psi + cos\phi \cdot cos\psi) \cdot v_b + (cos\phi \cdot sin\theta \cdot cos\psi + sin\phi \cdot sin\psi) \cdot w_b \tag{3.66}$$

$$w_v = -sin\theta \cdot u_b + sin\phi \cdot cos\theta \cdot v_b + cos\phi \cdot cos\theta \cdot w_b \tag{3.67}$$

$$n_z = -F_{Z_{aero}} / (m \cdot g_0) \tag{3.68}$$

$$n_y = -F_{Y_{aero}} /(m \cdot g_0) \quad (3.69)$$

$$M = V_T / a(h) \quad (3.70)$$

$$\gamma = \arcsin(cos\alpha \cdot cos\beta \cdot sin\theta - (sin\phi \cdot sin\beta + cos\phi \cdot sin\alpha \cdot cos\beta) \cdot cos\theta) \quad (3.71)$$

$$C_D = C_N \cdot sin\alpha + C_T cos\alpha \quad (3.72)$$

$$C_L = C_N \cdot cos\alpha - C_T sin\alpha \quad (3.73)$$

and $C_C, C_l, C_m, C_n, F_x, F_y, M_y$

3.2.3 Aero-data Model

The aircraft aero-data modeling consists of aero-data tables, interpolation routines and aero-data algorithms. This is a standard way of performing aerodynamic modeling. In these aero-data tables an interpolation is made and the six resulting aerodynamic coefficients, (C_T , C_N, C_C, C_l , Cm, Cn), are calculated. These coefficients are calculated with respect to a reference point as shown in figure 3.4. The aerodynamic reference point coincides with the nominal c.g. of the aircraft.

Figure (3.4) shows the definition of the direction of the forces and moments from the aero-data. The aerodynamic forces are given in the form of body fixed normal, tangential and side forces. The aerodynamic reference point (*OU*) and the centre of gravity (*OB*) are given in Figure (3.5). The reference point is fixed but the location of the c.g. can change. In the nominal case, these two points coincide. Deviation in c.g. from the aerodynamic reference point will give additional effects in the moment equations.

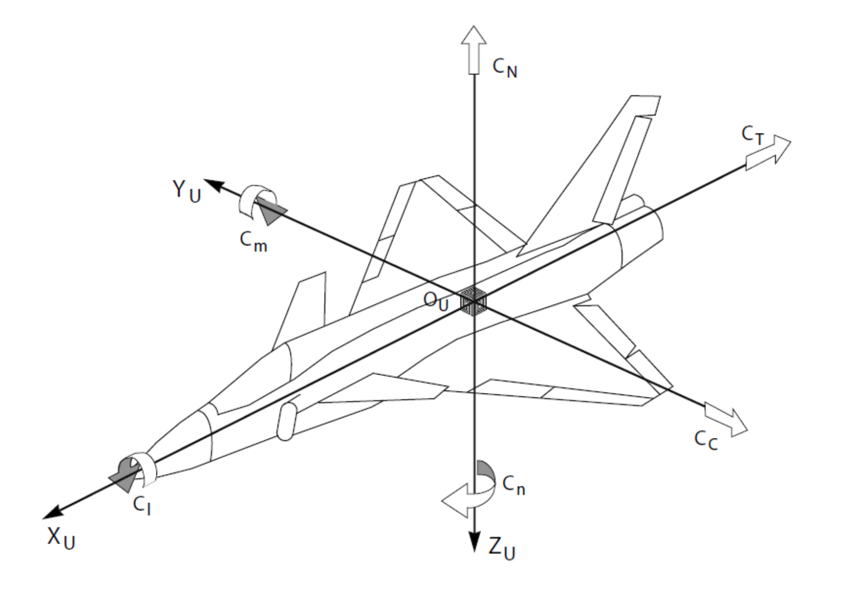


Figure 3.4: Definition of aerodynamic coefficients [9].

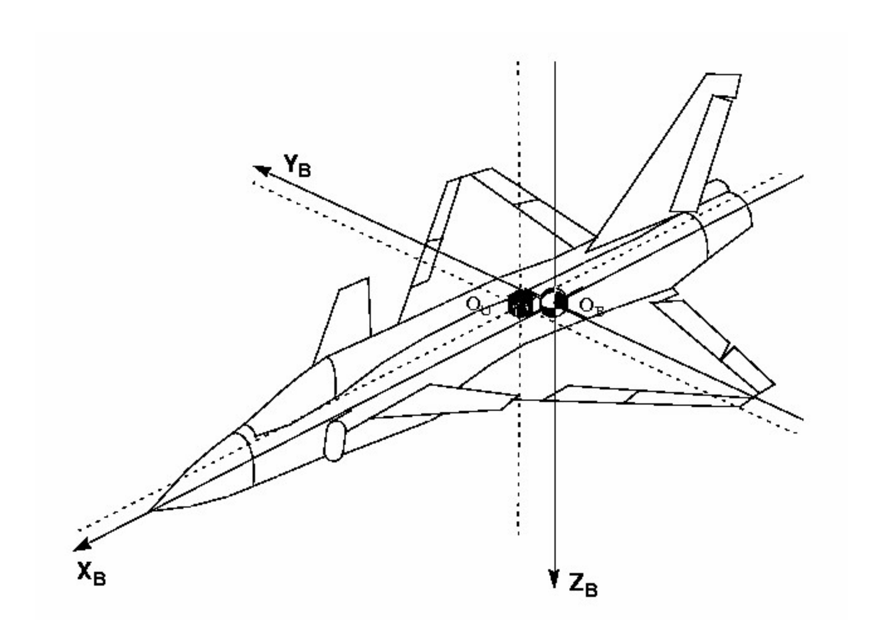


Figure 3.5: Definition of reference frames [9].

The aerodynamic model is built up in a conventional way, by interpolating (unstructured) data tables to obtain the different contributions. Different aero-data tables are used at different Mach numbers. A transition is made between Mach numbers 0.4 and 0.5 and at Mach number 1.4. The aero-data contains static aero-elastic effects and coupling

between lateral and longitudinal dynamics of the aircraft, i.e. C_{N_β} . The total aerodynamic forces and moments acting on the aircraft are calculated in the following way:

$$F_x = \bar{q}.S_{ref}.C_{T_{tot}} \quad (3.74)$$

$$F_y = \bar{q}.S_{ref}.C_{C_{tot}} \quad (3.75)$$

$$F_z = -\bar{q}.S_{ref}.C_{N_{tot}} \quad (3.76)$$

$$M_x = \bar{q}.S_{ref}.b_{ref}.C_{l_{tot}} - z_{cg}.F_y + y_{cg}.F_z \quad (3.77)$$

$$M_y = \bar{q}.S_{ref}.c_{ref}.C_{l_{tot}} - x_{cg}.F_z + z_{cg}.F_x \quad (3.78)$$

$$M_z = -\bar{q}.S_{ref}.b_{ref}.C_{l_{tot}} + x_{cg}.F_y - y_{cg}.F_x \quad (3.79)$$

Note that x_{cg}, y_{cg}, z_{cg} are not a fixed coordinate but they are the relative distance between the centre of gravity and the aerodynamic reference point.

3.2.4 Engine Model

The engine model contains data in two 2-dimensional tables describing the engine thrust. The two tables contain the available thrust from the engine, one with activated afterburner and the other without. The thrust is a function of the altitude and the Mach number, the input to the engine is the Throttle Stick Setting T_{ss}, which takes values between 0 and 1. When T_{ss} is greater than or equal to 0.8 the afterburner is active. There is a blending between the two tables smoothing the transition. The blending is done to make the model more smooth which simplifies the trimming.

Due to the time it takes to accelerate/decelerate the rotating parts of the engine, the dynamic response in the engine is modeled with a simple first-order lag filter.

$$T_{ss}(s) = \frac{0.5}{s+0.5}.T_{ss_{com}} \quad (3.80)$$

3.2.5 Actuators

The actuator model used is simply a first order transfer function with limited angular deflection and maximum angular rate. The time constant for the leading edge flap is chosen to have a different value compared with the other actuators. The available control actuators in the ADMIRE model are:

- Left canard (δ_{lc}) and right canard (δ_{rc})
- Left outer elevon (δ_{loe}) and left inner elevon (δ_{lie})
- Right inner elevon (δ_{rie}) and right outer elevon (δ_{roe})
- Rudder (δ_r)
- Airbrake (δ_{ab})

The sign of the actuator deflections follows the “right-hand-rule”. The “right-hand-rule” means that a positive deflection corresponds to a positive rotation assuming that the hinge line is parallel to the respective axis in the body-fixed reference frame [13].

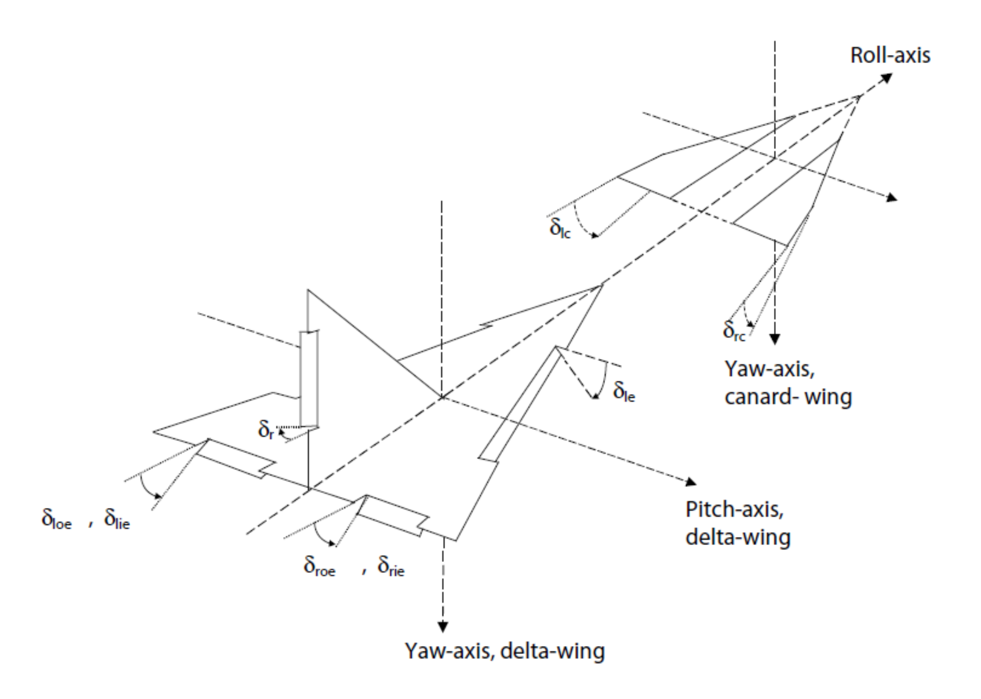


Figure 3.6: Definition of control surface deflection [9]

3.2.6 Sensors

The modeling of the sensors is identical to the models of sensors in the HIRM model [13]. In ADMIRE models of air data sensors(V_T, α, β, h), inertial sensors (p_b, q_b, r_b, n_z) , and attitude sensors (θ, ϕ) are implemented.

• Air data sensors:

$$\xi_{sensed}(s) = \frac{1}{1+0.02 \cdot s} \cdot \xi \tag{3.81}$$

Where $\xi = [V_T, \alpha, \beta, h]^T$

• Inertial sensors:

$$\xi_{sensed}(s) = \frac{1+0.005346 \cdot s+0.0001903 \cdot s^2}{1+0.03082 \cdot s+0.0004942 \cdot s^2} \cdot \xi \tag{3.82}$$

where $\xi = [p_b, q_b, r_b, n_z]^T$

• Attitude sensors:

$$\xi_{sensed}(s) = \frac{1}{1+0.0323 \cdot s+0.00104 \cdot s^2} \cdot \xi \tag{3.83}$$

where $\xi = [\theta, \phi]^T$

3.3 Flight Control System

Each aircraft should have its own flight envelope, which will depend on the individual vehicle are physical capabilities. Figure 3.7 shows a typical flight envelope for a supersonic aircraft, defined in terms of Mach number, covering velocity and air compressibility effects, and altitude to cover air temperature and density effects. The boundaries of the flight envelope are indicated by the physical limits shown in the Figure: the stall limit, at high incidence and low dynamic pressure, where the aircraft's wing lift is not sufficient to support the aircraft's weight; the performance limit due to the rarefaction of the atmosphere preventing a jet engine from sustaining its operation; the temperature limit due to the kinetic heating of the airframe by the viscous friction of the air; the loading limit, at high dynamic pressure, due to the aerodynamic loads acting on the airframe--a limit which is deliberately specified to provide an adequate margin against aircraft flutter. The forces and moments acting on the vehicle vary substantially across such an envelope. If we can control these forces and moments, then we have control of the aircraft's translational and rotational accelerations, and hence its velocities, attitude and position. The FCS aims to achieve this control via the aircraft's aerodynamic control surfaces and/or the thrust provided by the engines, which is normally controlled in its magnitude, but also, in some modern military aircraft, in its direction. The functions of an FCS can be structured into both primary functions, such as pitch, roll and yaw control, and secondary functions such as high lift, airbrake and lift-dump on the ground.

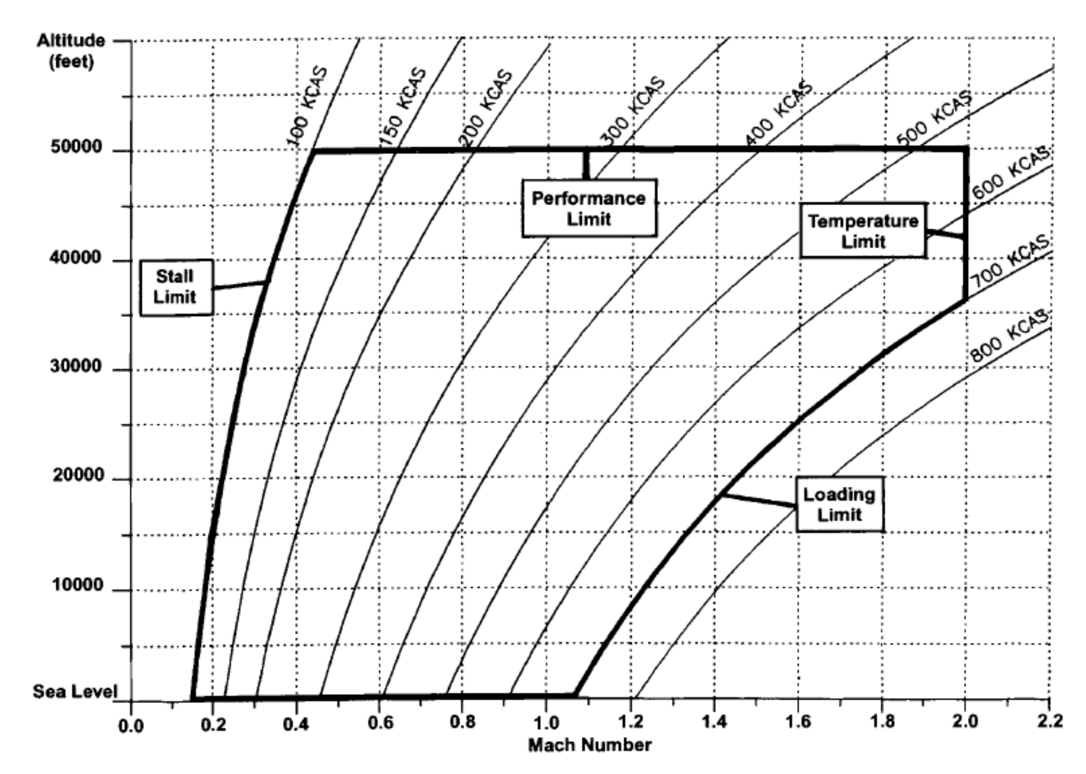


Figure 3.7: Example on aircraft flight envelope and its limitation [20].

A flight control system is an interconnected network of controls that allows a pilot to fly an aircraft. There are a number of different styles of flight control system, ranging from the basic to the extremely complex, and refinements to these systems are constantly being designed to make aircraft safer and easier to fly.

The earliest designs of flight control systems were purely mechanical. That means the system directly depends on the physically orders of the pilot and without any feedback, aircraft designers developed hydromechanical systems, which boosted the system with hydraulics to make it easier to operate. Hydromechanical systems included feedback system so that the pilot could not pull too hard or too suddenly and damage the plane.

Modern flight control systems utilize what is known as a fly-by-wire system. In these systems, the pilot uses electronic controls, which communicate with the various components of the system. Both analog and digital fly-by-wire systems are used on all types of aircraft military and civilian ones. Digital flight control systems give the designers the possibility to increase the safety factors by introducing multi redundancy system to perform the same task. Also such system allows the controls of a plane to respond to emerging issues before the pilot even notices them

ADMIRE use a simple FCS that provides basic stability and sufficient handling qualities within the operational envelope. Although the aerodynamic model envelope is valid up to 20 km, normal flight operation is significantly lower.

The FCS contains a longitudinal and a lateral part. The function of the longitudinal controller is pitch rate control ($q$com) below Mach number 0.58 and load factor control (nzcom) above Mach number 0.62. A blending function is used in the region in between, in order to switch between the two different modes. The longitudinal controller also contains a speed control (VTcom). The lateral controller enables the pilot to perform roll control around the velocity vector of the aircraft (pwcom) and to control the sideslip angle (βcom). The FCS is designed in 29 trim conditions using standard linear design methods.

The pilot control inceptors are longitudinal (Fes) and lateral (Fas) stick deflection, rudder pedal deflection (Frp) and throttle stick setting (Tss). For simplicity, linear stick gradients are used. The maximum longitudinal stick deflection is asymmetric, i.e. it is possible for the pilot to pull a larger command than to push.

The control selector (CS) is used to distribute the three control channels, *up*, *uq* and *u*β, out to the seven control actuators used by the FCS. A scheduling of the CS is done by using the Mach number and the altitude.

Since the flight controller is designed in a number of discrete points in the envelope, the gains in the FCS must be adjusted when the aircraft is operating between the initial design points. In ADMIRE all FCS gains and trim conditions are scheduled with the altitude and Mach number. In Figure (3.8) an example of a scheduled gain can be found.

In order to model the time delays present in an onboard computer implementation of the control laws, transport delays of 20 ms have been added to the actuators.

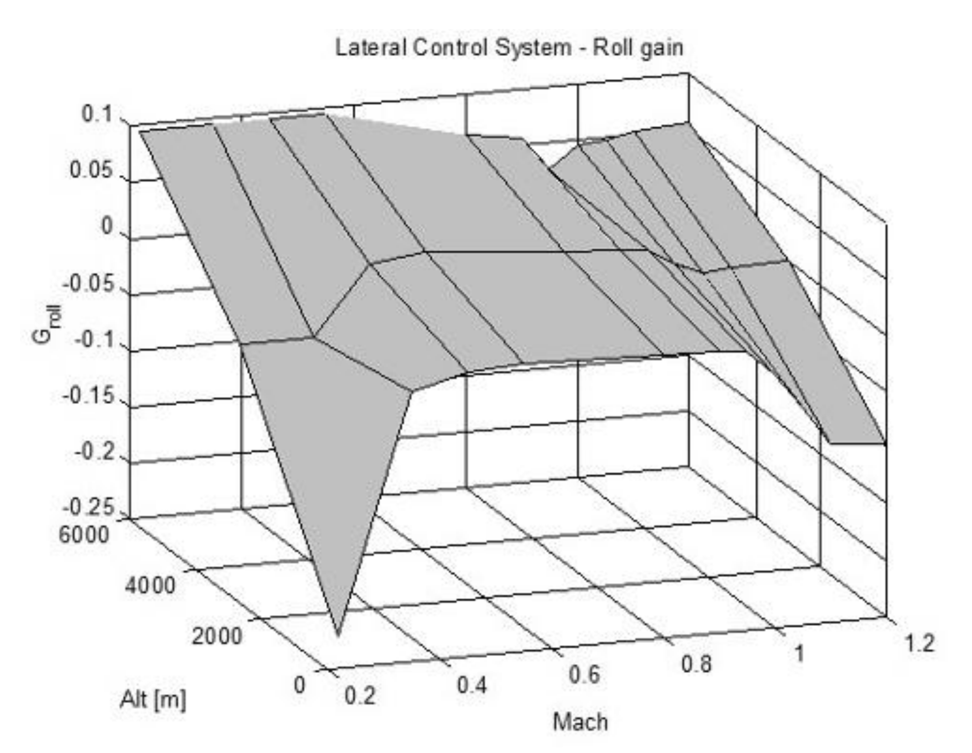


Figure. 3.8. Example of scheduled gain [9]

3.3.1 Longitudinal Controller

The longitudinal controller has two parts; a pitch and a speed controller. The speed controller is a gain and a lead filter. The controller maintains the commanded speed.

The pitch controller has two different modes of functionality. At lower speeds ($M < 0.58$) the function of the controller is to minimize the tracking error in the commanded pitch rate, which is generated by the pilot. At higher speeds ($M > 0.62$), the controller tracks the commanded load factor. For Mach numbers in between, a blending is performed.

The pitch rate controller consists of a stabilizing inner loop. The pitch rate and the angle of attack are fed back to the control selector and an outer loop where the tracking error, and the integrated value of it, is fed forward to the control selector. Due to the chosen design method only the deviation, values from the trimmed flight condition are fed to the gains. This means that the controller only commands deviation from the trimmed actuator settings. The trimmed values are added in front of the actuators. The purpose of the alpha-limiter is to provide a pitch rate command to the controller when the prescribed limit of angle of attack is exceeded. How this is done is described in [3].

The load factor controller has the same principal structure as the pitch rate controller, described above, except that nz is fed back in the outer loop and it does not contain any alpha-limiter. The longitudinal control laws were synthesized using Pole Placement Methods (PPM).

3.3.2 Lateral Controller

The lateral control system is a so-called lateral-directional control augmentation system (CAS). The body-axis roll rate is fed back to the ailerons to modify the roll-subsidence mode. Closed-loop control of roll rate is used to reduce the variation of roll performance with flight conditions. The calculation of the amplification factors are implemented in a C-function.

The inner feedback loop in the rudder channel provides roll damping by feeding back an approximation of the wind-axis yaw rate to the rudder. The wind-axis yaw rate is washed-out so that it operates only transiently and does not contribute to a control error when steady yaw rate is present. The yaw-rate feedback is equivalent to $\dot{\beta}$ feedback when φ and β are small.

When necessary the pilot can command a steady sideslip to the aircraft, because rudder inputs are applied via the rudder pedal gradient to the rudder actuator. The control system will tend to reject this disturbance input, so that the desirable effect of limiting the sideslip will be achieved.

The outer feedback loop in the rudder channel provides Dutch-roll damping through sideslip feedback to the rudder. The sideslip contributes to a control error and makes it possible to control the sideslip.

The cross-connection is known as the aileron-rudder interconnects (ARI). Its purpose is to provide the component of yaw rate necessary to achieve a wind-axis roll. The lateral control laws were also synthesized using PPM.

3.4 Chapter Summary

A brief introduction to aerodynamics was presented in this chapter. After that the used aircraft dynamic model in this thesis was presented in details based on the ADMIRE aircraft model that was developed by the Swedish Aeronautical Research Institute using aero data obtained from a generic single seated, single engine fighter aircraft with a delta canard configuration. The ADMIRE model include the actuator model, sensor model, and engine model.

Finally, the augmented flight control laws which used in ADMIRE also presented in addition to the flight envelop profile.

CHAPTER FOUR

CONTINUOUS GENETIC ALGORITHM

4.1 Introduction

Genetic Algorithms (GAs) are adaptive heuristic search algorithm premised on the evolutionary ideas of natural selection and genetics. The basic concept of GAs is designed to simulate processes in natural system necessary for evolution, specifically those that follow the principles first laid down by Charles Darwin of the survival of the fittest. As such, they represent an intelligent exploitation of a random search within a defined search space to solve a problem.

First pioneered by John Holland in the 1960s, Genetic Algorithms have been widely studied, experimented, and applied in many fields in engineering worlds. Not only do GAs provide an alternative method to solving problem, but also outperforms other traditional methods in most of the problems. Many of the real world problems have proven difficult for traditional methods but ideal for GAs.

The GA procedure is based on the Darwinian principle of survival of the fittest. An initial population is created containing a predefined number of individuals (or solutions), incorporating the variable information. Each individual has an associated fitness measure, typically representing an objective value. The concept that fittest (or best) individuals in a population will produce fitter offspring is then implemented in order to reproduce the next population. Selected individuals are chosen for crossover at each generation, with an appropriate mutation factor to randomly modify the genes of an individual, in order to develop the new population. The result is another set of individuals based on the original

population leading to subsequent populations with better (min. or max.) individual fitness. Therefore, the algorithm identifies the individuals with the high fitness values, and those with lower fitness will naturally be discarded from the population.

Genetic algorithms can be distinguished from calculus-based and enumerative methods for optimization by the following characteristics [1,5,12]:

- GAs search for optimal solution using a population of individuals, not a single individual. This very important characteristic gives GAs much of their search power and points to their parallel nature.
- GAs use only objective function information. No other auxiliary information is required. Much of the interest in genetic algorithms is due to the fact that they belong to the class of efficient domain-independent search strategies that are usually superior in performance to traditional methods without the need to incorporate highly domain-specific knowledge.
- GAs use probabilistic transition rules, not deterministic rules in contrast with the calculus based and enumerative methods.

The population-based nature of genetic algorithms gives them two major advantages over other optimization techniques. First, it identifies the parallel behavior of genetic algorithms that is realized by a population of simultaneously moving search individuals or candidate solutions [2]. Implementation of GAs on parallel machines, which significantly reduces the CPU time required, is a major interesting benefit of their implicit parallel nature. Second, information concerning different regions of solution space is passed actively between the individuals by the crossover procedure. This information exchange makes genetic algorithm an efficient and robust method for optimization,

particularly for the optimization of functions of many variables and nonlinear functions. On the other hand, the population-based natures of genetic algorithms also result in two drawbacks. First, more memory space is occupied; that is, instead of using one search vector for the solution, N_p search vectors are used which represent the population size. Second, GAs normally suffers from computational burden when applied on sequential machines.

The fact that GAs use only objective function information without the need to incorporate highly domain-specific knowledge, points to both the simplicity of the approach from one side and its versatility from the other. This means that once a genetic algorithm is developed to handle a certain problem, it can easily be modified to handle other types of problems by changing the objective function in the existing algorithm. This is why genetic algorithms are classified as general-purpose search strategies.

The stochastic behavior of genetic algorithms cannot be ignored as a main part that gives them much of their search efficiency. GAs employ random processes to explore a response surface for a specific optimization problem. The advantage of this type of behavior is the ability to escape from local extrema without supervision [2].

4.2 Conventional Genetic Algorithm

Conventional genetic algorithm generally consists of the following steps:

1) ***Initialization*****:** An initial population comprising of N_p individuals is generated in this phase at the genotype level by filling the bit strings randomly by 1 or 0 values. The coding process is then used to find phenotype values of the population.

2) ***Evaluation*****:** The fitness, a nonnegative measure of quality used as a measure to reflect the degree of goodness of the individual, is calculated for each individual in

the population according to its phenotype structure. The fitness is a function of the performance index, and both equality and inequality constraints.

3) ***Selection*:** In the selection process, individuals are chosen from the current population to enter a mating pool devoted to the creation of new individuals for the next generation such that the chance of a given individual to be selected to mate is proportional to its relative fitness. This means that best individuals receive more copies in subsequent generations so that their desirable traits may be passed onto their offspring. This step ensures that the overall quality of the population increases from one generation to the next.

4) ***Crossover:*** Crossover provides the means by which valuable information is shared among the population. It combines the features of two parent individuals to form two children individuals that may have new phenotype structures compared to those of their parents and plays a central role in GAs.

Conventional crossover involves exchanging genes (bits) between each pair of parents selected from the mating pool. It is generally applied with a relatively high probability of crossover, P_c. Three known schemes are generally used which include the single-point, the multi-point and the uniform crossover schemes. In single-point crossover method, a crossover point is randomly selected along the parent strings and the crossover operator exchanges the characters after the crossover point between the two-selected parent strings. In multi-point crossover method, m different random crossover points across the chromosome are selected first. Then, the genotype string of each chromosome is spitted into $m+1$ part. The offspring are created by choosing genotype fragments from each parent alternately; that is, swapping partial strings with the same size and position between the two

selected chromosomes. In uniform crossover method, each chromosome position (gene) is crossed with some probability, which is typically one-half. That is, each corresponding pair of genes exchanges their values independently with probability of 0.5. As a result, a random crossing mask is implicitly generated with the probability of one at any position typically being set to one-half. Characters from the parental strings having ones at the corresponding positions in the crossing mask are swapped while generating the offspring strings, and the remaining characters remain intact.

5) ***Mutation:*** Mutation is often introduced to guard against premature convergence. Generally, over a period of several generations, the gene pool tends to become more and more homogeneous. The purpose of mutation is to introduce occasional perturbations to the variables to maintain the diversity within the population. In conventional mutation operator, the bitwise complement mutation is applied at the gene level with some low probability of mutation, P_m. It is realized by performing bit inversion (flipping) on some randomly selected bit positions of offspring bit strings.

6) ***Replacement:*** After generating the offspring population through the application of the genetic operators to the parent population, the parent population is totally or partially replaced by the offspring population depending on the replacement scheme used. This completes the "life cycle" of the population.

7) ***Termination:*** GA is terminated when some convergence criterion is met. The fact that conventional genetic algorithm uses both genotype and phenotype data presentation requires some coding process that relates both schemes. If the optimization problem consists of N_t variables and each variable is represented by a

substring of N_s characters or genes, then chromosomes or individuals are formed by cascading the genes of N_t variables forming a longer string of length $L=N_tN_s$ genes. In this way, the population may be viewed as a vector of N_p elements where each element consists of L genes. The coding process requires the user to specify the desired accuracy or resolution to be used, which specifies the number of genes per substring. In addition to that, the lower and upper bounds of the variables are also required. The decoding process is governed by the following equation:

$$x = x_{lower} + \frac{x_{dec}}{2^{N_s} - 1}(x_{upper} - x_{lower}) \qquad (4.1)$$

where

x represents the value of the variable

x_{dec} represents the decimal decoded value of the variable

x_{lower} is the lower bound of the variable

x_{upper} is the upper bound of the variable.

To summarize the evolution process in conventional genetic algorithm, an individual is a candidate solution of the variables to be optimized; that is, each individual consists of a string of $L=N_tN_s$ genes. Initially, N_p individuals are randomly generated representing the initial population. The population is sorted and ranked depending on the fitness value. The population undergoes the selection process, which results in a mating pool among which pairs of individuals are crossed with probability P_c. This process results in an offspring generation where every individual child undergoes mutation with probability P_m. After that, the next generation is produced according to the replacement strategy applied. This process is repeated till the convergence criterion is met where the N_t variables of the best individual are the

required unknown values. The block diagram of the genetic algorithm is given in Figure (4.1).

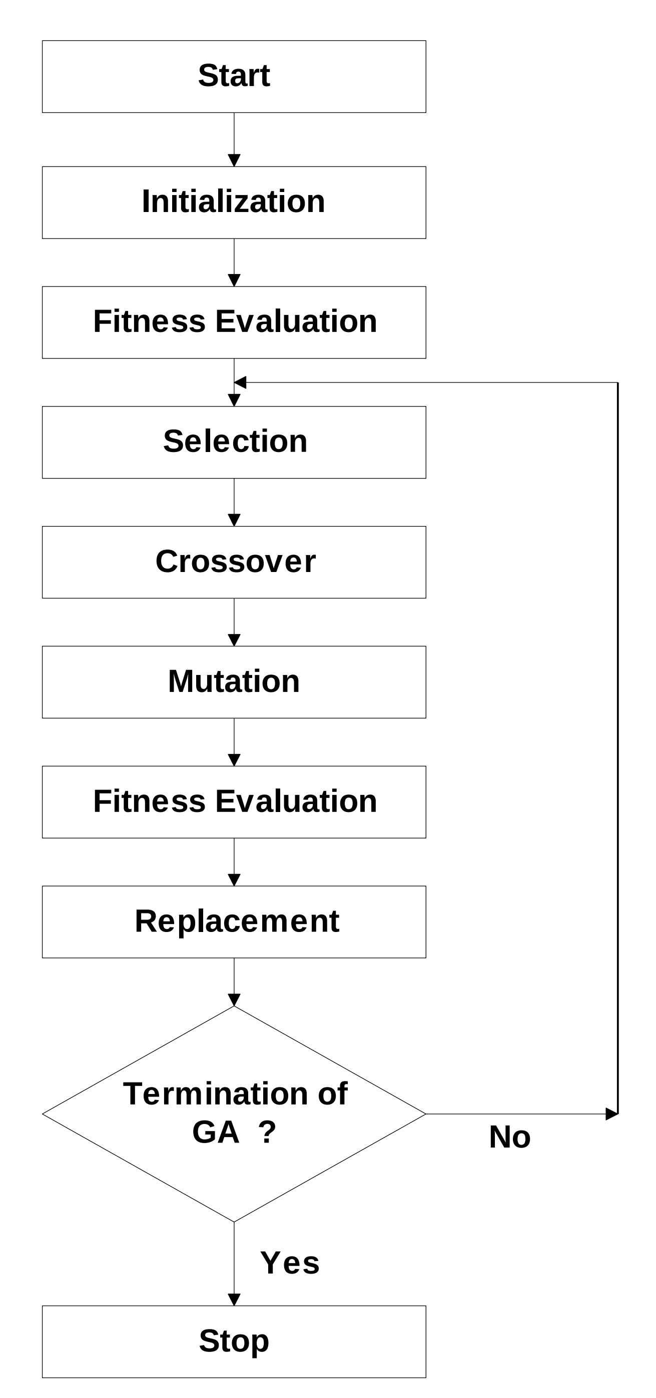


Figure 4.1: Block diagram for a typical Genetic Algorithm.

4.3 Continuous Genetic Algorithm

A continuous genetic algorithm (CGA) was shown to be an efficient method for the solution of optimization problems in which the parameters to be optimized are correlated with each other or the smoothness of the solution curve should be achieved. It has been successfully applied in the motion planning of robot manipulators in the field of robotics and in the numerical solution of boundary value problems in the field of applied mathematics. Their novel development has opened the doors for wide applications of the algorithm.

Before going to the detailed description of CGA, the conditions about the continuous functions that can be used in such algorithm should be clearly stated. In relation to the initialization function, any smooth function that is close enough to the expected solution curve can be used. It is to be noted that the closer the initialization function to the final solution, the faster the convergence speed since the coarse tuning stage of CGA in this case will be bypassed and the algorithm will jump to the fine tuning stage. In the case that there is no prior information about the expected solution curves, then any smooth function can be used and a mixture of functions will be beneficial in this case to result in a diverse initial population. The effect of the initial population usually dies after few tens of generations and the convergence speed after that is governed by the selection mechanism, crossover and mutation operators [1,2].

Regarding the crossover function, it should be within the range [0, 1] such that the offspring solution curve will start with the solution curve of the first parent and gradually change their values until they reach the solution curve of the second parent at the other end.

The mutation function may be any continuous function. However, both functions should satisfy any problem-specific constraints if such constraints exist.

The continuous genetic algorithm consists of the following steps [1,2]:

1) ***Initialization:*** In this phase, an initial population comprising of Np smooth individuals is randomly generated. In this work, two smooth functions are used for initializing the population: the modified Gaussian function and the modified Sinc function. The modified Gaussian function is given as follows:

$$P_j(h,i) = u_{left}(h) + \frac{u_{right}(h) - u_{left}(h)}{N_t - 1}(i-1) + A * \exp\left[\frac{-(i-\mu)^2}{2\sigma^2}\right] \quad (4.2)$$

While the modified Sinc function is governed by the equation:

$$P_j(h,i) = u_{left}(h) + \frac{u_{right}(h) - u_{left}(h)}{N_t - 1}(i-1) + \mathrm{sinc}(x_s - i) * i \quad (4.3)$$

for all $1 \le h \le N_c, \; 1 \le i \le N_t$

where

$P_j(h,i)$ is the i^{th} node value of the h^{th} curve for the j^{th} parent

$u_{left}(h)$ is the value of the leftmost variable of the h^{th} curve

$u_{right}(h)$ is the value of the rightmost variable of the h^{th} curve

$u_{left}(h)$ *and* $u_{right}(h)$ are randomly generated subject to the problem constraints

N_t is the total number of nodes along each solution curve

A represents a random number within the range $[-3R(h), 3R(h)]$, with $R(h)=| u_{right}(h)- u_{left}(h)|$

μ is a random number within the range $[1, N_t]$

σ is a random number within the range $[1, N_t/6]$

$x_s = f(x_v, N_t)$, index series at which to interpolate.

x_v, random number within the range *[1-5]*.

For both functions, μ specifies the center of the function, while σ specifies its degree of dispersion. The two initialization functions are shown in figure 4.2, and figure 4.3.

2) ***Evaluation*****:** The fitness is calculated for each individual in the population. It is a function of the performance index, in the clearance process we try to found the maximum value of the load factor n_z or the maximum value of angle of attack.

$$\begin{Bmatrix} \text{Fitness} = \max_{p\in P} n_{z_{max}}(p) \\ \text{Fitness} = \max_{p\in P} \alpha_{max}(p) \end{Bmatrix} \tag{4.4}$$

3) ***Selection*****:** Individuals from the current population which are selected to mate according to their relative fitness so that best individuals receive more copies in subsequent generations.

4) ***Crossover:*** Crossover is the way through which information is shared among the population. The crossover process combines the features of two parent individuals, say *j* and *k*, to form two children individuals, say *L* and *L+1*, as given by the equations:

$$C_L(h,i) = W(h,i)P_j(h,i) + [1 - W(h,i)]P_k(h,i) \tag{4.5}$$

$$C_{L+1}(h,i) = [1 - W(h,i)]P_j(h,i) + W(h,i)P_k(h,i) \tag{4.6}$$

$$W(h,i) = 0.5\left[1 + tanh\left(\frac{i-\mu}{\sigma}\right)\right] \tag{4.7}$$

For all $1 \le h \le N_c$, $1 \le i \le N_t$

where

P_j and P_k represent the two parents chosen from the mating pool,

C_L and C_{L+1} are the two children obtained through crossover process,

W represents the crossover weighting function within the range [0, 1],

μ is a random number within the range [1, N_t],

σ is a random number within the range [1, $N_t /6$].

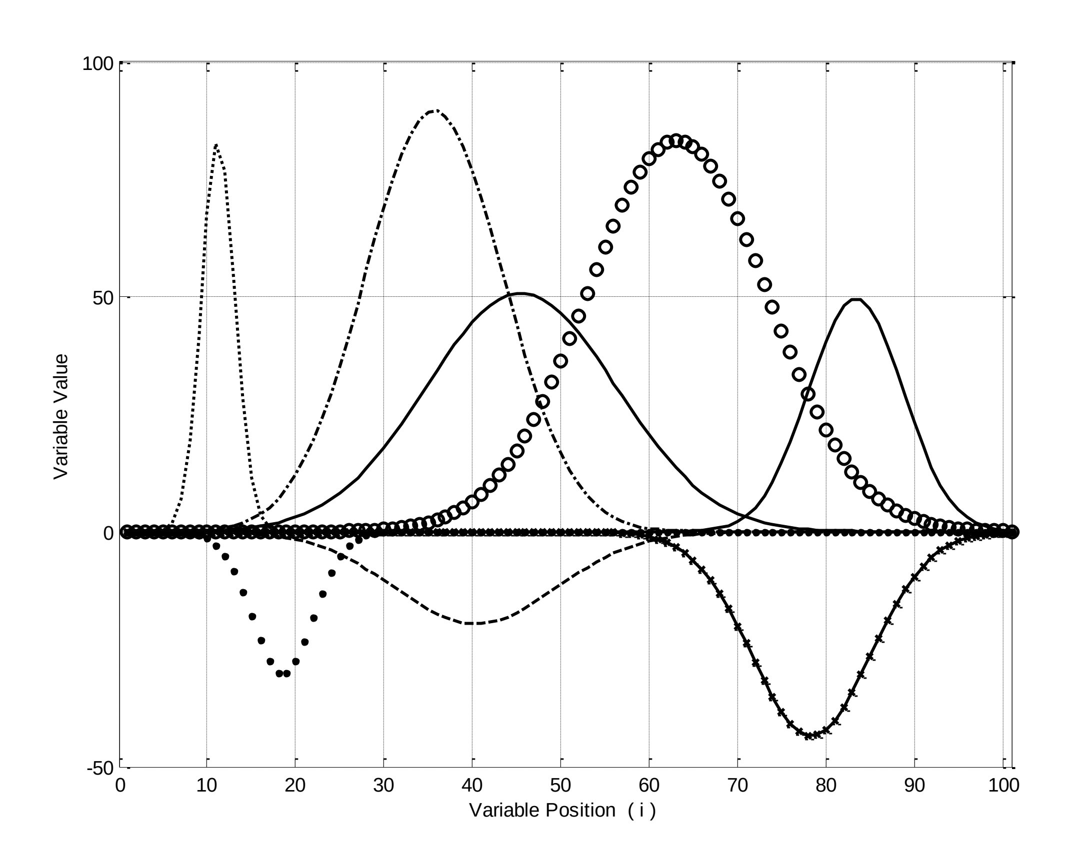


Figure 4.2: modified Gauss function, the initialization functions used in the continuous genetic algorithm.

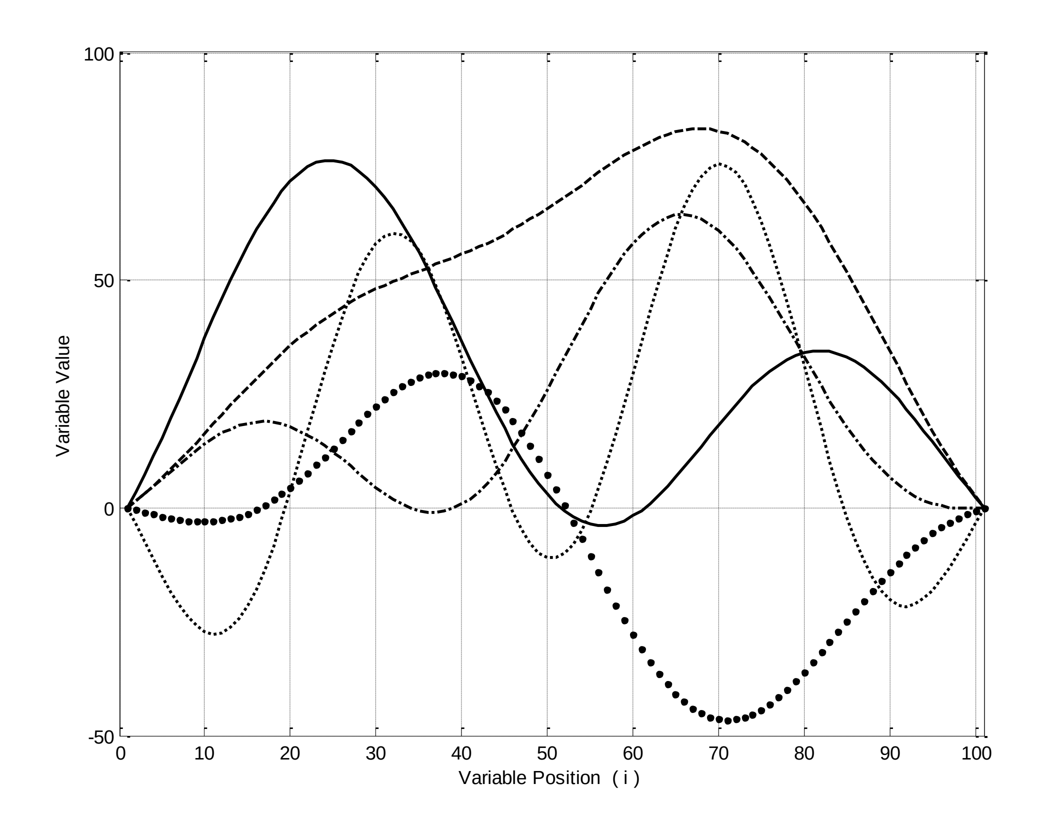


Figure 4.3: modified Sinc function, the initialization functions used in the continuous genetic algorithm.

The crossover operator in CGA is applied with a slight modification to that used in conventional GA. In the CGA, pairs of individuals are crossed with probability P_{ci}. Within the pair of parents that should undergo crossover process, individual curves are crossed with probability P_{cc}. That is, the h^{th} smooth curve of the first parent is crossed with the h^{th} smooth curve of the second parent with P_{cc} probability. If P_{ci} value is set to 0.5 and P_{cc} value is set to 0.5, then one pair of parents between two pairs is likely to be crossed, and within that pair, $N_c/2$ of the curves are likely to be crossed. Figure (4.4) shows the crossover process between two random solution curves. It is clear that new information is incorporated in the children while maintaining the smoothness of the resulting solution curves.

5) ***Mutation*****:** The function of mutation is to introduce occasional perturbations to the variables to maintain the diversity within the population. The mutation process in CGA is governed by the following formulas[14]:

$$C_j^{\,m}(h,i) = C_j(h,i) + d \times M(h,i) \qquad (4.8)$$

$$M(h,i) = exp\left[\frac{-(i-\mu)^2}{2\sigma^2}\right] \qquad (4.9)$$

where : C_j represents the j^{th} child produced through the crossover process,

C_j^m is the mutated j^{th} child,

M is the Gaussian mutation function,

d represents a random number within the range [-*range(h), range(h)*], with *range(h)* representing the difference between the minimum and maximum values of the h^{th} smooth curve of child C_j ,

μ and σ are as given in the crossover process.

In mutation process, each individual child undergoes mutation with probability P_{mi}. However, for each child that should undergo mutation process, individual curves are mutated with probability P_{mc}. If P_{mi} value is set to 0.5 and P_{mc} value is set to 0.5, then one child out of two children is likely to be mutated, and within that child, $N_c/2$ of the solution curves are likely to be mutated.

Figure (4.5) shows the mutation process in a solution curve of a certain child. As in crossover process, some new information is incorporated in the children while maintaining the smoothness of the resulting solution curves.

6) ***Replacement*****:** In this step, the parent population is totally or partially replaced by the offspring population depending on the replacement scheme used. This completes the "life cycle" of population.

7) ***Termination*****:** CGA is terminated when some convergence criterion is met. Possible convergence criteria are: the fitness of the best individual so far found exceeds a threshold value, the maximum number of generations is reached, or the progress limit (the improvement in the fitness value of the best member of the population over a specified number of generations being less than some predefined threshold) is reached. After terminating the algorithm, the optimal solution of the problem is the best individual so far found.

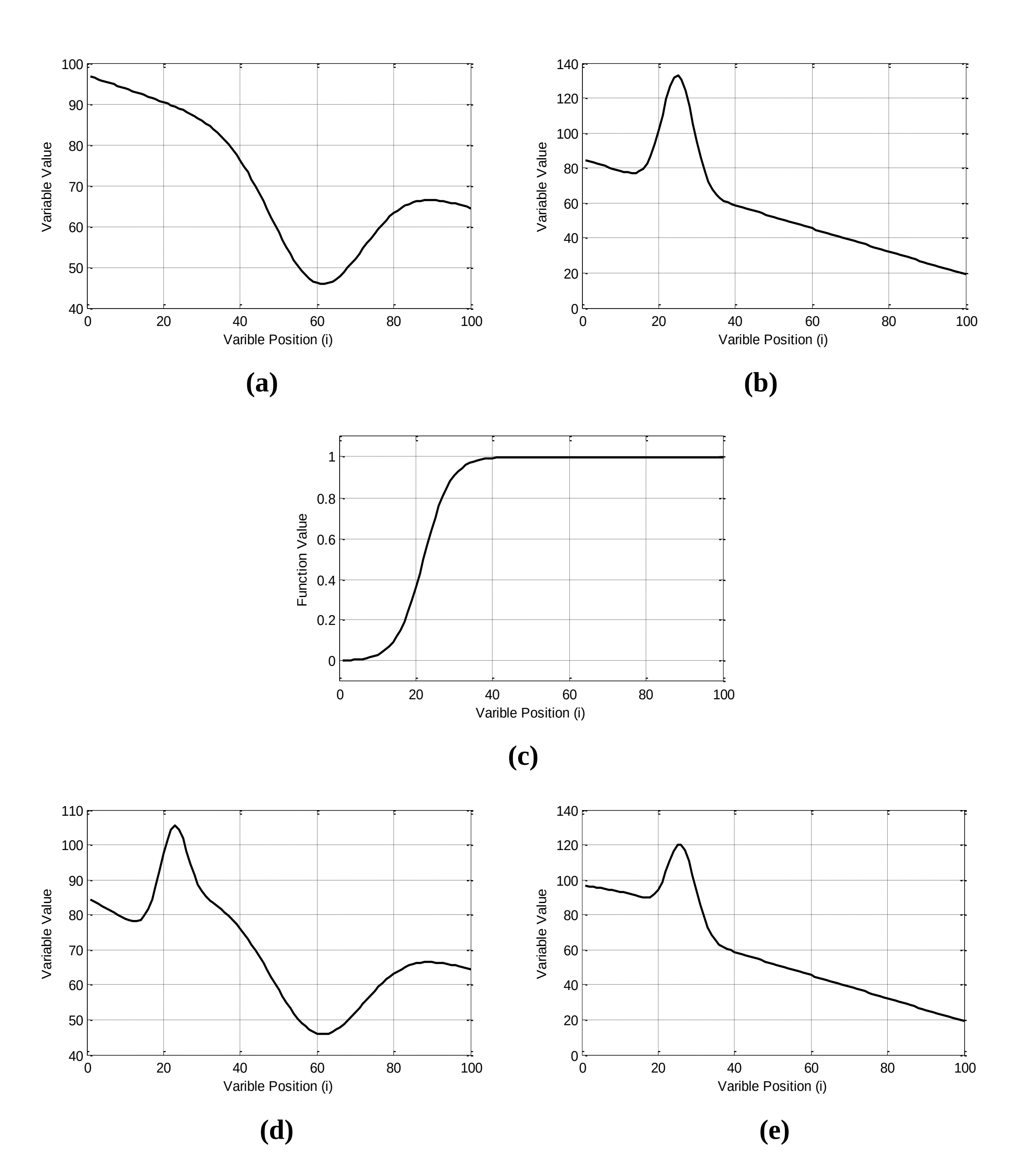


Figure 4.4: Crossover process (a) first parent, (b) second parent, (c) crossover function, (d) first child, (e) second child.

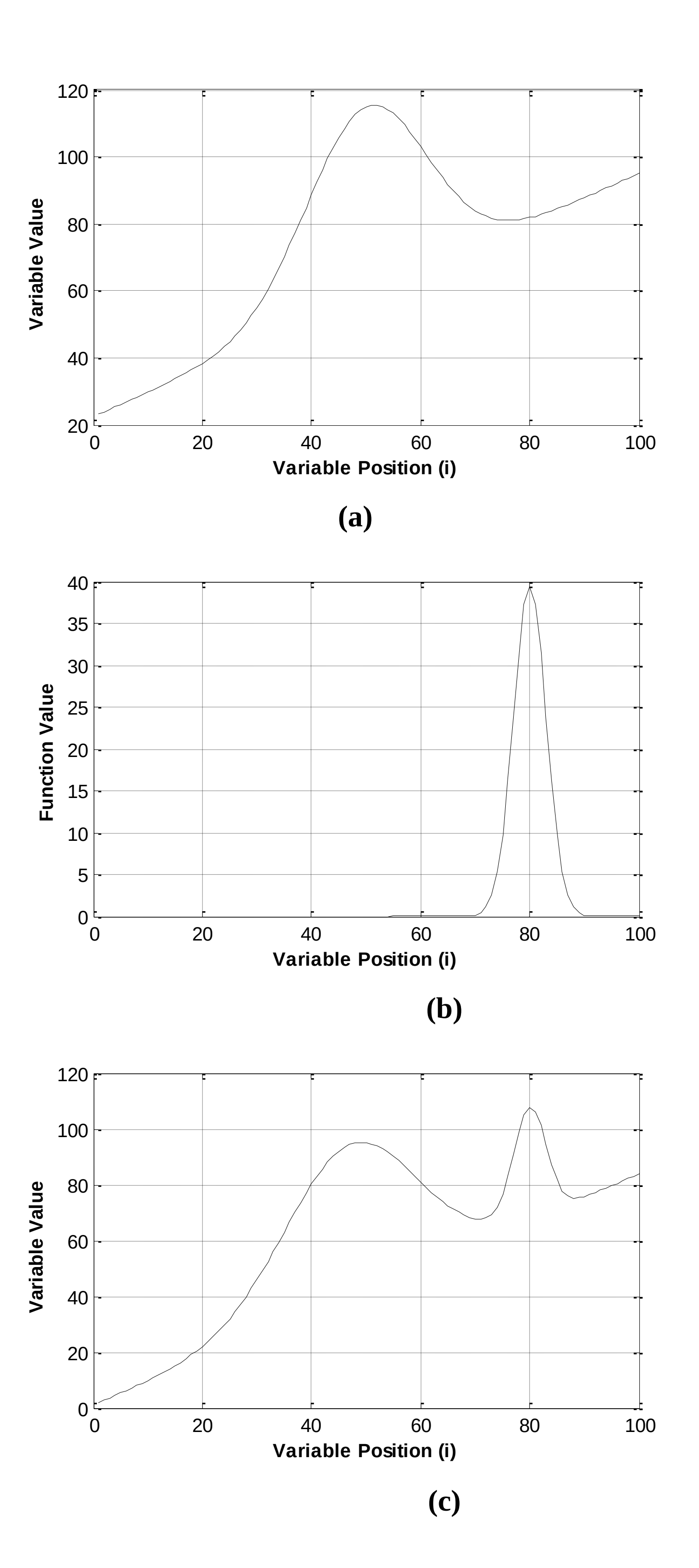


Figure 4.5: Mutation process (a) offspring, (b) mutation function, (c) offspring after mutation.

To summarize the evolution process in CGA, an individual is a candidate solution of the required solution curves; that is, each individual consists of N_c solution curves each consisting of N_t variables. This results in a two-dimensional array of the size $N_c \times N_t$. The population undergoes the selection process, which results in a mating pool among which pairs of individuals are crossed with probability P_{ci}. Within that pair of parents, individual solution curves are crossed over with probability P_{cc}. This process results in an offspring generation where every individual child undergoes mutation with probability P_{mi}. Within that child, individual solution curves are mutated with probability P_{mc}. After that, the next generation is produced according to the replacement strategy applied.

This process is repeated until the convergence criterion is met where the N_c solution curves of the best individual are the required solution curves. The final goal of discovering the required solution curves is translated into finding the fittest individual in genetic terms.

In addition to the previous operators, the following operators and/or enhancement schemes are introduced to enhance the performance of the algorithm:

a) ***Elitism:*** The preservation of the best solution or solutions and moving it or them to the next generation is vital to the effectiveness of GA [7]. It is utilized to ensure that the fitness of the best candidate solution in the current population is larger than or equal to that in the previous population. In other words, it guarantees that the best fitness in the population is a monotically non-decreasing function.

b) ***Extinction and immigration operator:*** This operator is applied when all individuals in the population are identical or when the improvement in the fitness value of the best individual over a certain number of generations is less

than some threshold value. The number of individuals in the population associated with better fitness grows exponentially. Therefore, after some generations, the mating pool will consist of almost identical members. GA thus tends to stagnate; "extinction and immigration" operator is used to bypass this difficulty.

This operator, as indicated by its name, consists of two stages; the first stage is the extinction process where all of the individuals in the current generation are removed except the best-of-generation individual. The second stage is the mass immigration process where the extinct population is filled out again by generating "N_p-*1*" individuals to keep the population size fixed. The generated population is divided into two equal segments each of $N_p/2$ size; the first segment, with $2 \le j \le N_p/2$, is generated as in the initialization phase, while the other segment is generated by performing continuous mutations to the best-of-generation individual as given by the formula:

$$P_j(h,i) = P_1(h,i) + d \times M(h,i), \left(\frac{N_p}{2}+1\right) \le j \le N_p \quad (4.10)$$

where

P_j is the j^{th} parent generated using immigration operator,

P_1 represents the best-of-generation individual,

M is the Gaussian mutation function,

d represents a random number within the range [-*range(h)*, *range(h)*], with *range(h)* representing the difference between the minimum and maximum values of the h^{th} solution curve of the best individual.

c) ***Bounds of the variables:*** In some optimization problems, the variables to be optimized should remain within some specific bounds. In order to achieve this goal, a regular

check should be done on the values of the variables such that they will not exceed the given limits during the evolution process. To be more specific, the initialization and mutation processes should have this regular check on variables values.

Regarding the initialization process, this check for parents is required if the modified Gaussian function is used due to any possible overshoot or undershoot that might go beyond the limits. While in the case of tangent hyperbolic function, this step is not required. For the mutation process, the check for variable values of the offspring is required because these values might exceed the limits after the addition/subtraction of the mutation function. Regarding the crossover process, if a regular check is performed in the initialization and mutation processes then this step is not required in the crossover process due to the nature of the crossover function.

For this purpose, two methods are used in a work that maintains the smoothness of the solution curves: the fixed limit method and the mirror limit method. The mirror limit method is governed by the Equation (4.11) and (4.12) [1]:

$$P_j(h,i)=\begin{Bmatrix} P_j(h,i); & U_{\min}(h)\le P_j(h,i)\le U_{\max}(h) \\ U_{\max}(h)-\left[P_j(h,i)-U_{\max}(h)\right]; & P_j(h,i)>U_{\max}(h) \\ U_{\min}(h)+\left[U_{\min}(h)-P_j(h,i)\right]; & P_j(h,i)<U_{\min}(h) \end{Bmatrix} \tag{4.11}$$

The fixed limit method, on the other hand, is governed by the following equation:

$$P_j(h,i)=\begin{Bmatrix} P_j(h,i); & U_{\min}(h)\le P_j(h,i)\le U_{\max}(h) \\ U_{\max}(h); & P_j(h,i)>U_{\max}(h) \\ U_{\min}(h); & P_j(h,i)<U_{\min}(h) \end{Bmatrix} \tag{4.12}$$

for all $1 \le h \le N_c$, $1 \le i \le N_t$

where

P_j is the j^{th} parent in case of initialization phase or the j^{th} child in case of mutation process,

$U_{min}(h)$ represents the lower bound of the h^{th} solution curve,

$U_{max}(h)$ represents the upper bound of the h^{th} solution curve.

The two methods are illustrated in Figure (4.6). As stated previously, the two methods result in smooth solution curve.

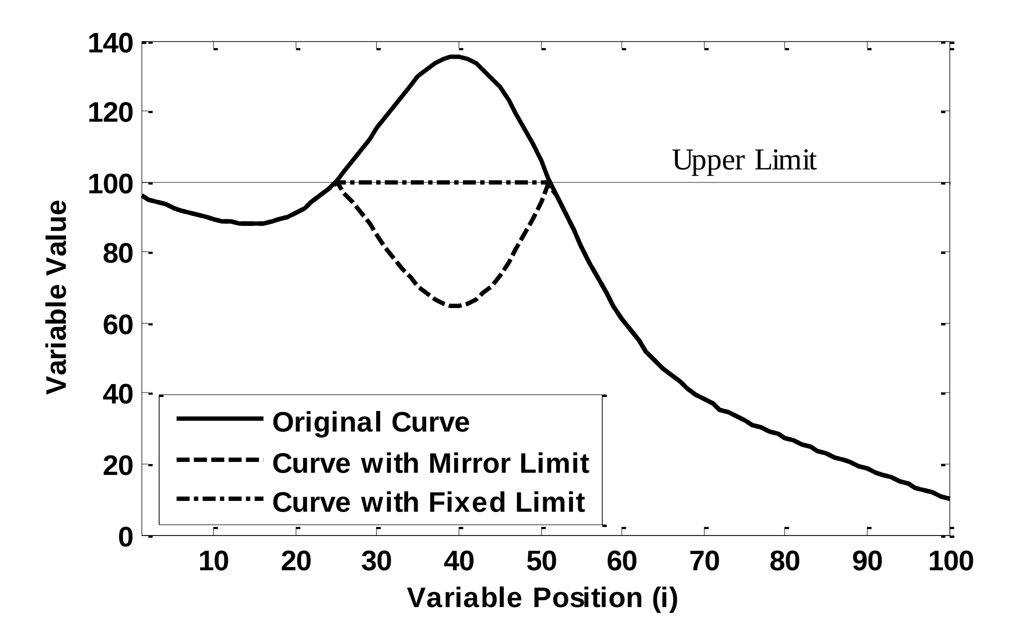


Figure 4.6: Used methods to limit variables values in continuous genetic algorithm.

4.4 Implementation:

The input data to the algorithm is divided into two main parts: the CGA related parameters and the flight control clearance optimization problem related parameters. The CGA related parameters include the population size, N_p; the initialization function; the selection scheme used; the replacement method; the individual crossover probability, P_{ci}; the curve crossover probability, P_{cc}; the individual mutation probability, P_{mi}; the curve mutation probability, P_{mc}; the method used when the solution curves exceed the given limit; the immigration threshold value and the corresponding number of generations; and finally, the termination criterion.

The flight control clearance optimization problem related parameters include the number of solution curves, N_c; the number of nodes along the solution curve, N_t; the variables limits for each solution curve ($U_{min}(h)$ and $U_{max}(h)$ for $h=1,\ldots, N_c$); the leftmost and the rightmost variables of the h^{th} solution curve ($u_{left}(h)$ and $u_{right}(h)$ for $h=1,\ldots, N_c$).

In order to increase the efficiency of the CGA for such optimization problem, which has nonlinear and coupled states some changes done on the CGA. These changes begun by using the new initialization function the modified Sinc function instead of the tanch function.

In order to increase the diversity of the generated population, immediate after any population generation process the new population were subject to crossover process.

The selection scheme, which used here in CGA, is the ranked-based scheme. Rank-based selection chooses a prescribed number of parent individuals with the highest fitness according to the rank-based ratio, R_{rb}, and performs the mating process by choosing parents at random from this subpopulation of the size $R_{rb}N_p$.

Stochastic universal selection guarantees that the number of copies of an individual selected is almost proportional to its fitness, which is not necessarily the case for roulette wheel selection. In half biased selection, one mate is selected as in roulette wheel selection, while the other mate is selected randomly from the original population.

The size of the two populations is almost equal since the offspring generation replaces the parent generation completely except when elitism is used to copy the best individual or individuals from the parent population. In this case, if ρ elite parents are copied, then the size of the offspring population will be either "N_p-ρ" members (generational replacement type-I) if the ρ elite members enter the population directly, or N_p members (generational replacement type-II) if the worst ρ members of the offspring population are deleted before the ρ elite parents join the population forming a new population of N_p individuals which wholly replaces the parents (i.e., leaving the population size constant throughout the evolution).

The next population is formed as follow:

- Elite parents were passed direct to the next generation.
- The offspring individuals produced using crossover between the best parents based on randomly selected individual.
- The offspring individuals produced using mutation for previous individuals.
- New generated individual using the initialization function.

However, the proposed schemes are used to maintain sufficient diversity in the next population, and to improve the efficiency of the CGA.

4.5 Chapter Summary

In this chapter, a detailed description of the continuous genetic algorithm as well as the conventional genetic algorithm has been carried out. The evaluation, selection, replacement and the termination steps are identical in both algorithms. The difference between the two algorithms lies in the way the initialization phase, the crossover operator and the mutation operator are applied while maintaining the same goals. These operators are applied at the curve level in case of continuous genetic algorithm, while they are applied at the variable level in case of conventional genetic algorithm. The flight control clearance optimization problem handled by CGA involve the determination of the optimal set of pilot command input that maximize the angle of attack (AOA)/load factor.

CHAPTER FIVE

SIMULATION AND RESULTS

5.1 Introduction

The CGA was used to generate the pilot command input signals for both Pitch pilot command (Force on Elevon Stick, Fes) and Roll pilot command (Force on Aileron Stick, Fas). These two inputs were used during the simulation of ADMIRE model to find the aircraft response which yields the thirty-two states describe in chapter three. In this work, the interest is on only two of these states which are the load factor and angle of attack which were extracted from the aircraft response. These two states, the load factor and angle of attack, were used as fitness function in GCA. As it discussed earlier, the CGA was used to generate the needed initial population of pilot command input based on the Load Factor and Angle of Attack. The CGA evolution process, as shown in figure (5.1), was repeated until specified limiting criteria were met as explained in chapter four.

All of the calculations in the CGA were implemented and programmed using MATLAB platform. The algorithm was coupled with ADMIRE aircraft model that was developed by the Swedish Aeronautical Research Institute using aero data obtained from a generic single seated, single engine fighter aircraft with a delta canard configuration.

The results of the clearance of flight control laws based on load factor criteria and angle of attack criteria using CGA and comparison with the results of conventional genetic algorithms are presented and discussed in the sequel.

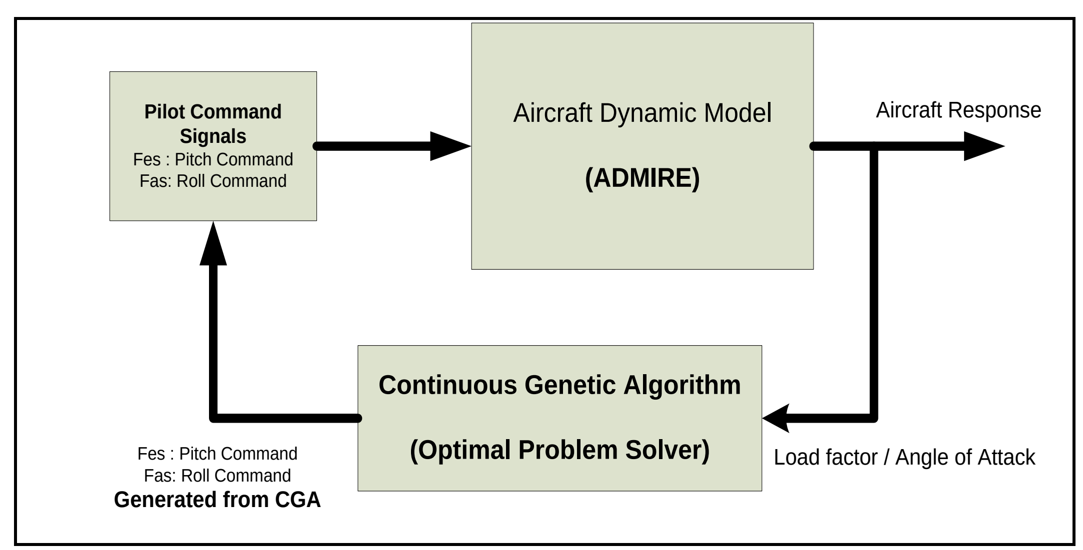


Figure 5.1: Block diagram for the clearance process

5.2 CGA parameters setup

The initial setting of the CGA parameters used in this work and explained in chapter four are as follows: the population size was set to 64 individuals, the crossover probability was set to 0.5, and the mutation probability was set to 0.5. The generational replacement scheme was modified to include the elite parent, crossed over parents, and mutated parent in addition to the new initializes parents.

During the iteration, unlike other GA methods, the initialization of new population was carried out and was made to share the offspring population by 12.5% of the population size. The crossover process settings, used in this work and based on previous experience, allowed 25% of the off spring population to be passed to the next generation. The rank-based selection strategy is used where the rank-based ratio is set to 10%.

This offspring population schemes is formed as follow:

- Elite parents were passed direct to the next generation.

- The offspring individuals produced using crossover between the best parents based on randomly selected individual.
- The offspring individuals produced using mutation for previous individuals.
- New generated individual using the initialization function.

However, the proposed schemes are used to maintain sufficient diversity in the next population, and to improve the efficiency of the CGA. CGA become more efficient and fast. This scheme is proposed based on the previous experience gained during using CGA to solve different types of optimization problem.

Immigration operator is applied when the improvement in the fitness value of the best individual of the population over 20 generation is less than 1e-6. The CGA is stopped when one of the following conditions is met. First, number of immigration process taking place more than 100 times; and second when the generation number exceeds 400. Due to the stochastic nature of CGA, four runs were performed for every point in the flight envelope. Statistical analysis based in Xbar charts are presented in appendix A for the Load Factor based results.

5.3 The selected points in the flight envelope for clearance

Table (1) shows the selected point in the flight envelope that used in the clearance process. Several researchers to validate the used optimization algorithm used these points. The same selected points are shown in Figure (5.2). The aircraft model trimmed to the selected altitude and speed value after that the pilot command inputs generated from the

CGA algorithm applied to aircraft model for 10 second. The load factor and angle of attack extracted from the aircraft model response were used after that in CGA as fitness function.

The simulation was performed using two initialization functions which are the modified Gauss function and the modified Sinc function. The results obtained using the above set parameter values are presented and discussed in the following sections.

Table 5.1: The selected operating points in the flight envelope.

Point #	Speed [Mach]	Altitude [m]
1	0.3	1000
2	0.6	6000
3	0.8	3000
4	1	5000
5	1.2	3000
6	1.2	6000

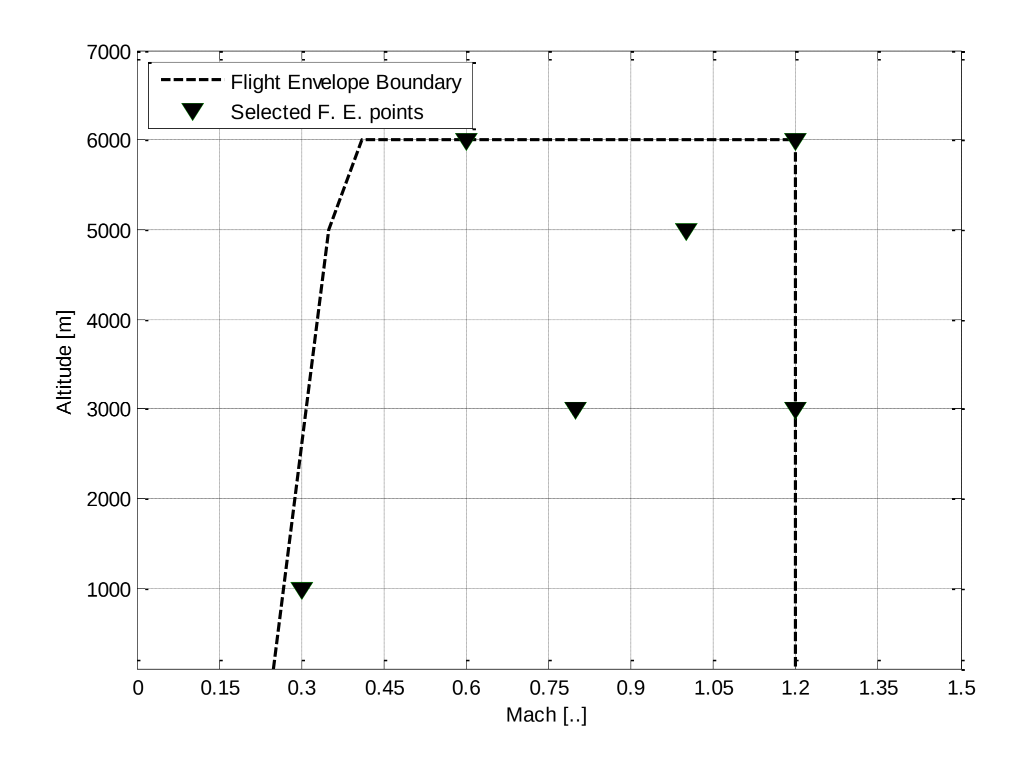


Figure 5.2: The selected operating points in flight envelope

5.4 Clearance of flight control laws based on Load Factor Criteria exceedance

The selected points in the flight envelope were processed and solved to find the maximum value for the load factor which can be obtained due to the pilot command inputs. In order to see the affect of the population size; the population size which used in CGA are 64 and 128 individuals

Tables (5.2-5) show the results obtained from this simulation. The number of iteration (generation) that required to obtain the maximum load factor and the maximum load factor value are presented in the tables for each selected point in the flight envelope. Figures (7-12) show the convergence performance of CGA for the selected points in the flight envelope. Each of these figures shows the convergence performance for the cases that results in using the two initialization modified Gauss and modified Sinc functions, and two deferent values for the population size, 64 and 128 individuals. Pitch and roll pilot command obtained from CGA which represent the pilot command that generate the maximum load factor (n_{zmax}) value are shown in Figures (13-18). These results show that the maximum value for the load factor is 10.9g obtained at 3000 m and 0.8 Mach speed. This means the load factor criteria exceed the limit (-3g - 9g) specified by the clearance process [3]. This result agrees with that obtained by other investigators using other optimization methods.

Table 5.2. The load factor simulation results for deferent run using modified gauss function and population size 64 individuals.

Mach	Alt	Run 1		Run 2		Run 3		Run 4		MAX Value
		Iter. No.	nz	Iter. No.	nz	Iter. No.	nz	Iter. No.	nz	
0.3	1000	158	10.558	189	10.420	319	10.569	138	10.426	10.569
0.6	6000	368	10.404	133	10.443	169	10.473	256	10.420	10.473
0.8	3000	213	10.642	301	10.399	257	10.712	179	10.618	**10.712**
1	5000	202	10.657	154	10.529	320	10.571	176	10.509	10.657
1.2	3000	189	10.425	401	10.449	188	10.664	147	10.301	10.664
1.2	6000	198	10.484	180	10.546	152	10.461	137	10.485	10.546

Table 5.3 The load factor simulation results for deferent run using modified sinc function and population size 64 individuals.

Mach	Alt	Run 1		Run 2		Run 3		Run 4		MAX Value
		Iter. No.	nz	Iter. No.	nz	Iter. No.	nz	Iter. No.	nz	
0.3	1000	160	10.650	295	10.828	313	10.612	171	10.562	10.828
0.6	6000	142	10.480	275	10.721	250	10.812	177	10.517	10.812
0.8	3000	269	10.600	401	10.569	195	10.408	137	10.535	10.600
1	5000	155	10.425	225	10.540	143	10.625	246	10.606	10.625
1.2	3000	215	10.741	262	10.639	245	10.668	214	10.900	**10.900**
1.2	6000	194	10.380	184	10.543	209	10.735	310	10.663	10.735

Table 5.4: The load factor simulation results for deferent run using using modified gauss function and population size 128 individuals.

Mach	Alt	Run 1		Run 2		Run 3		Run 4		MAX Value
		Iter. No.	nz	Iter. No.	nz	Iter. No.	nz	Iter. No.	nz	
0.3	1000	143	10.508	183	10.268	200	10.436	208	10.469	10.508
0.6	6000	139	10.470	381	10.496	207	10.348	195	10.661	10.661
0.8	3000	285	10.829	293	10.653	228	10.809	256	10.839	**10.839**
1	5000	180	10.688	202	10.657	196	10.4816	239	10.511	10.688
1.2	3000	189	10.425	401	10.449	188	10.664	147	10.301	10.664
1.2	6000	152	10.553	168	10.459	247	10.828	190	10.440	10.828

Table 5.5: The load factor simulation results for deferent run using modified sinc function and the population size 128 individuals.

Mach	Alt	Run 1		Run 2		Run 3		Run 4		MAX Value
		Iter. No.	nz	Iter. No.	nz	Iter. No.	nz	Iter. No.	nz	
0.3	1000	275	10.636	299	10.583	204	10.813	401	10.786	10.813
0.6	6000	255	10.595	237	10.730	216	10.842	210	10.503	**10.842**
0.8	3000	189	10.425	401	10.449	188	10.664	147	10.301	10.664
1	5000	401	10.614	223	10.459	211	10.311	294	10.563	10.614
1.2	3000	122	10.481	148	10.638	278	10.804	136	10.514	10.804
1.2	6000	235	10.600	204	10.716	280	10.655	381	10.709	10.716

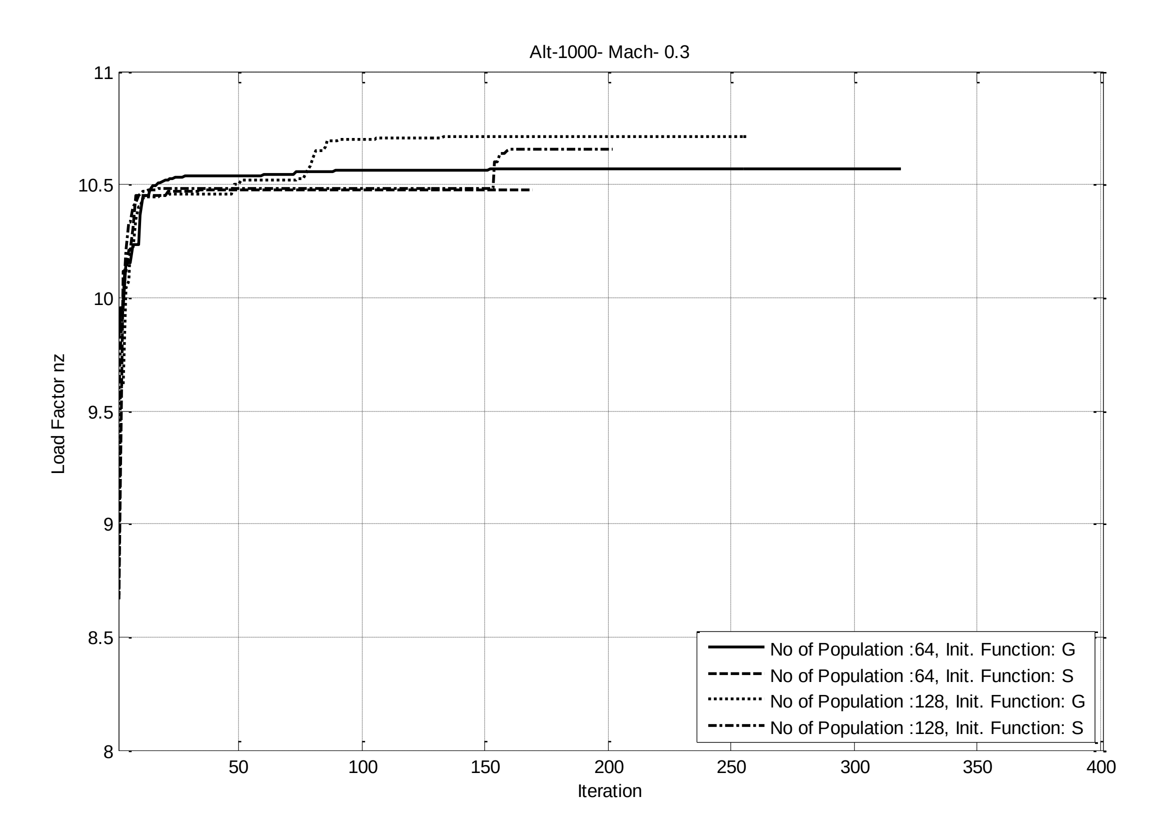


Figure 5.3: Maximum load factor (n_{zmax}) versus the iteration (Generation). Result using CGA with population size, 64 and 128, and both initialization function, modified gauss function (G) and modified sinc function (S), aircraft model trimmed at 1000 m altitude and 0.3 mach speed.

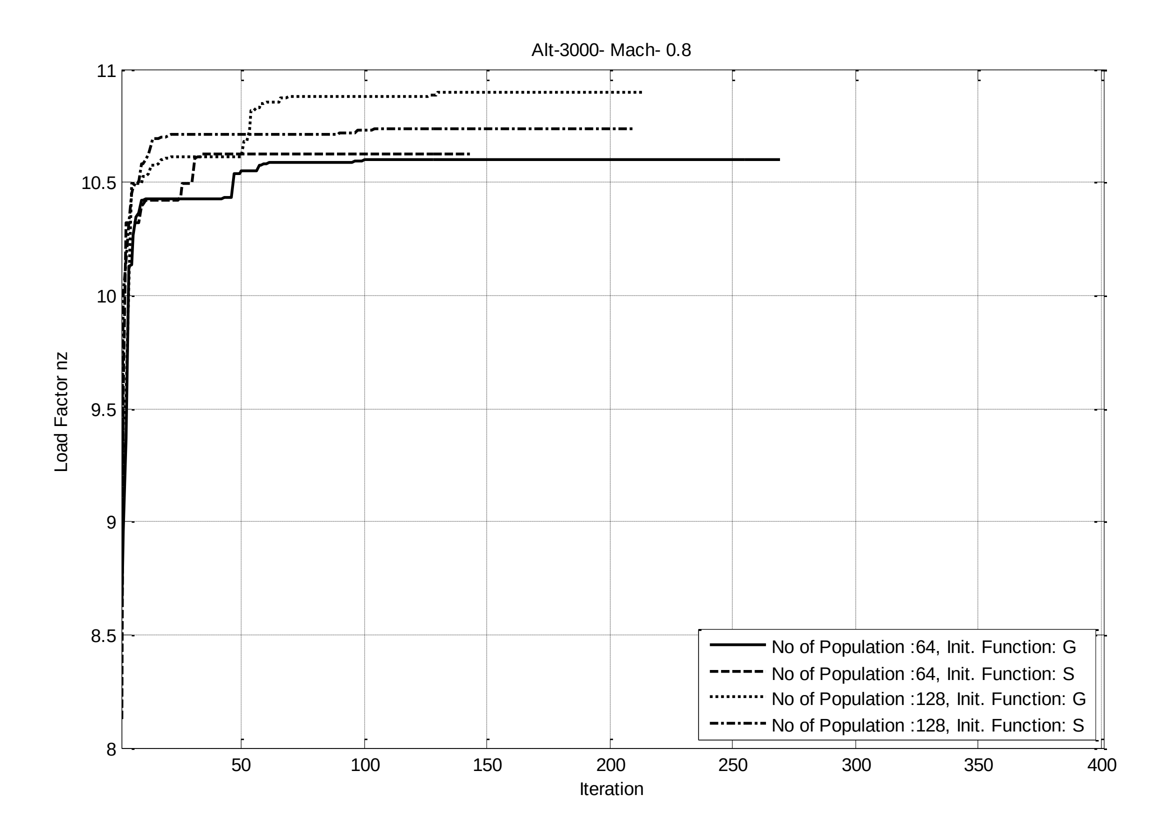


Figure 5.4: Maximum load factor (nzmax) versus the iteration (Generation). Result using CGA with population size, 64 and 128, and both initialization function, modified gauss function (G) and modified sinc function (S), aircraft model trimmed at 3000 m altitude and 0.8 mach speed.

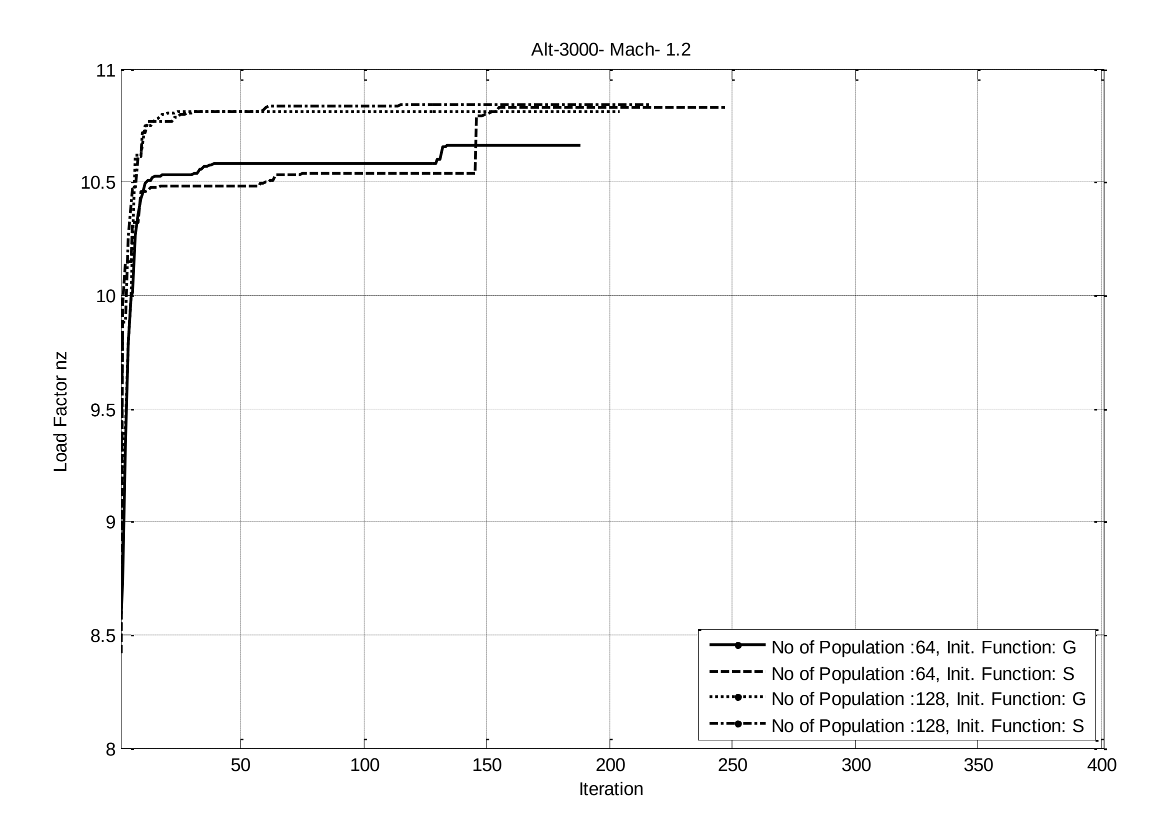


Figure 5.5: Maximum load factor (nzmax) versus the iteration (Generation). Result using CGA with population size, 64 and 128, and both initialization function, modified gauss function (G) and modified sinc function (S), aircraft model trimmed at 3000 m altitude and 1.2 mach speed.

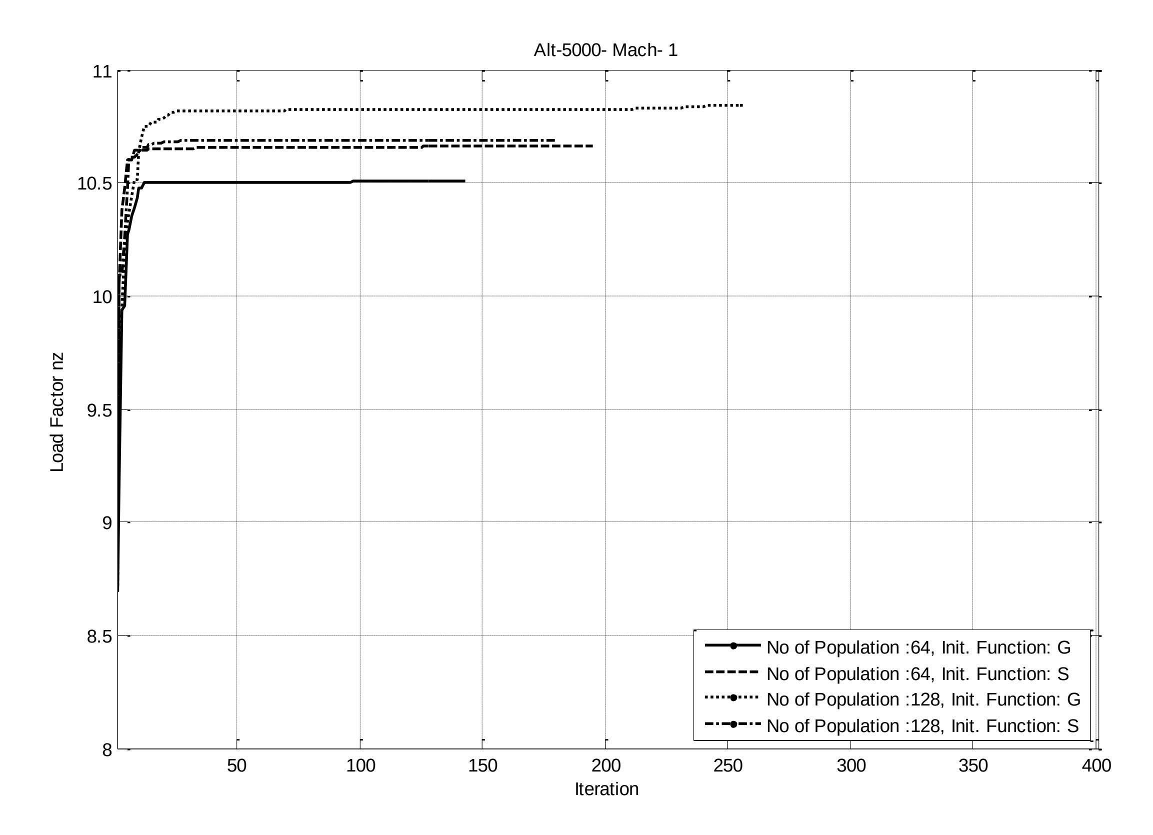


Figure 5.6: Maximum load factor (nzmax) versus the iteration (Generation). Result using CGA with population size, 64 and 128, and both initialization function, modified gauss function (G) and modified sinc function (S), aircraft model trimmed at 5000 m altitude and 1.0 mach speed.

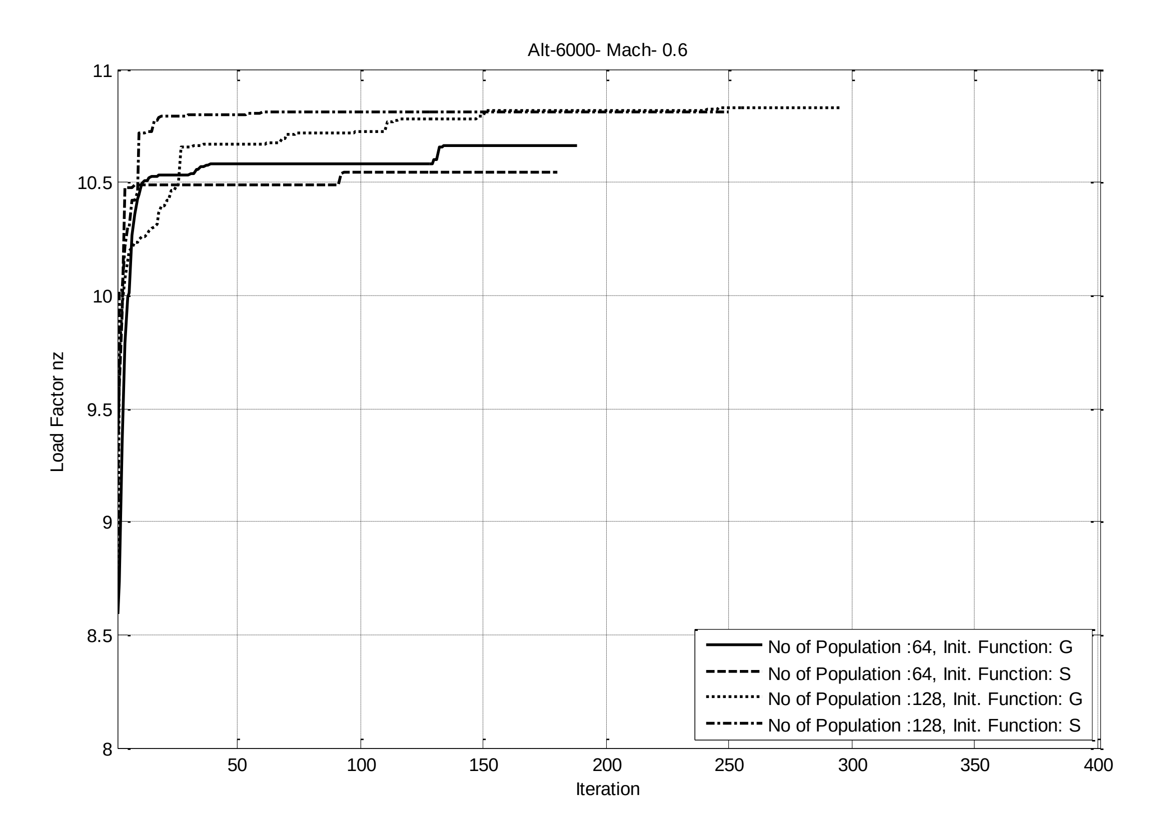


Figure 5.7: Maximum load factor (nzmax) versus the iteration (Generation). Result using CGA with population size, 64 and 128, and both initialization function, modified gauss function (G) and modified sinc function (S), aircraft model trimmed at 6000 m altitude and 0.6 mach speed.

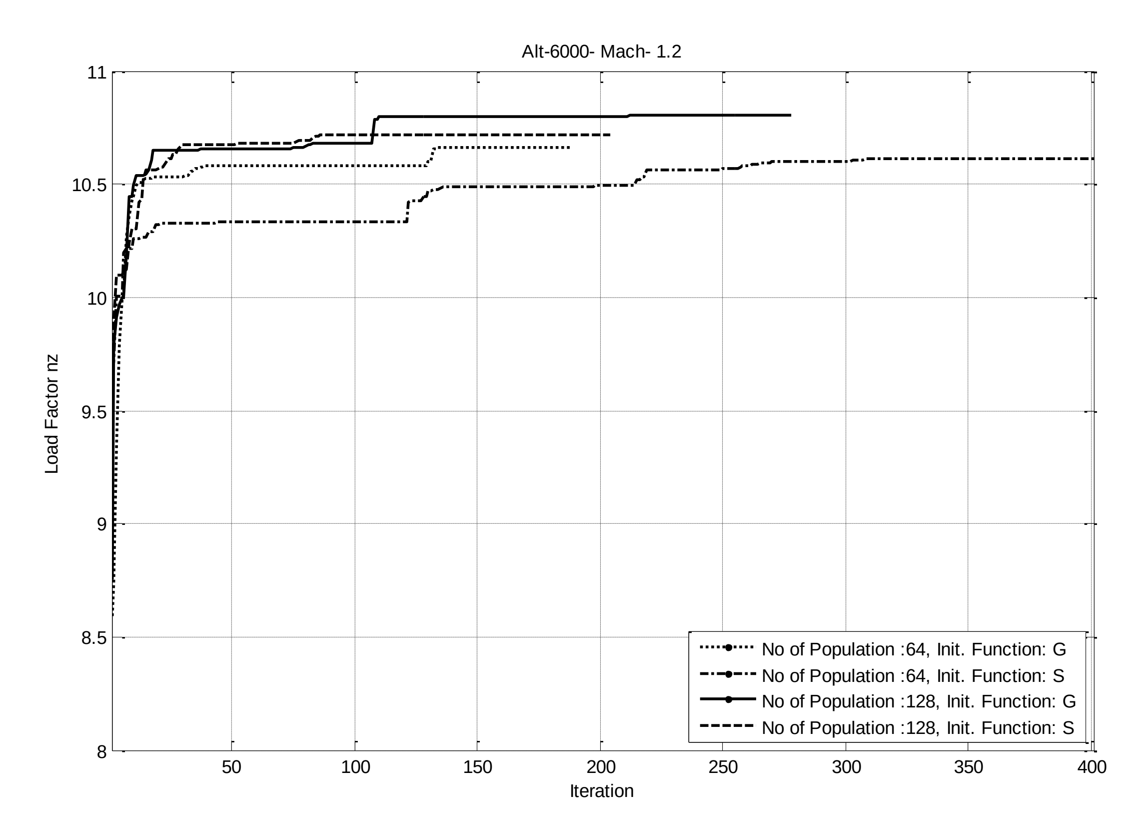


Figure 5.8: Maximum load factor (nzmax) versus the iteration (Generation). Result using CGA with population size, 64 and 128, and both initialization function, modified gauss function (G) and modified sinc function (S), aircraft model trimmed at 6000 m altitude and 1.2 mach speed.

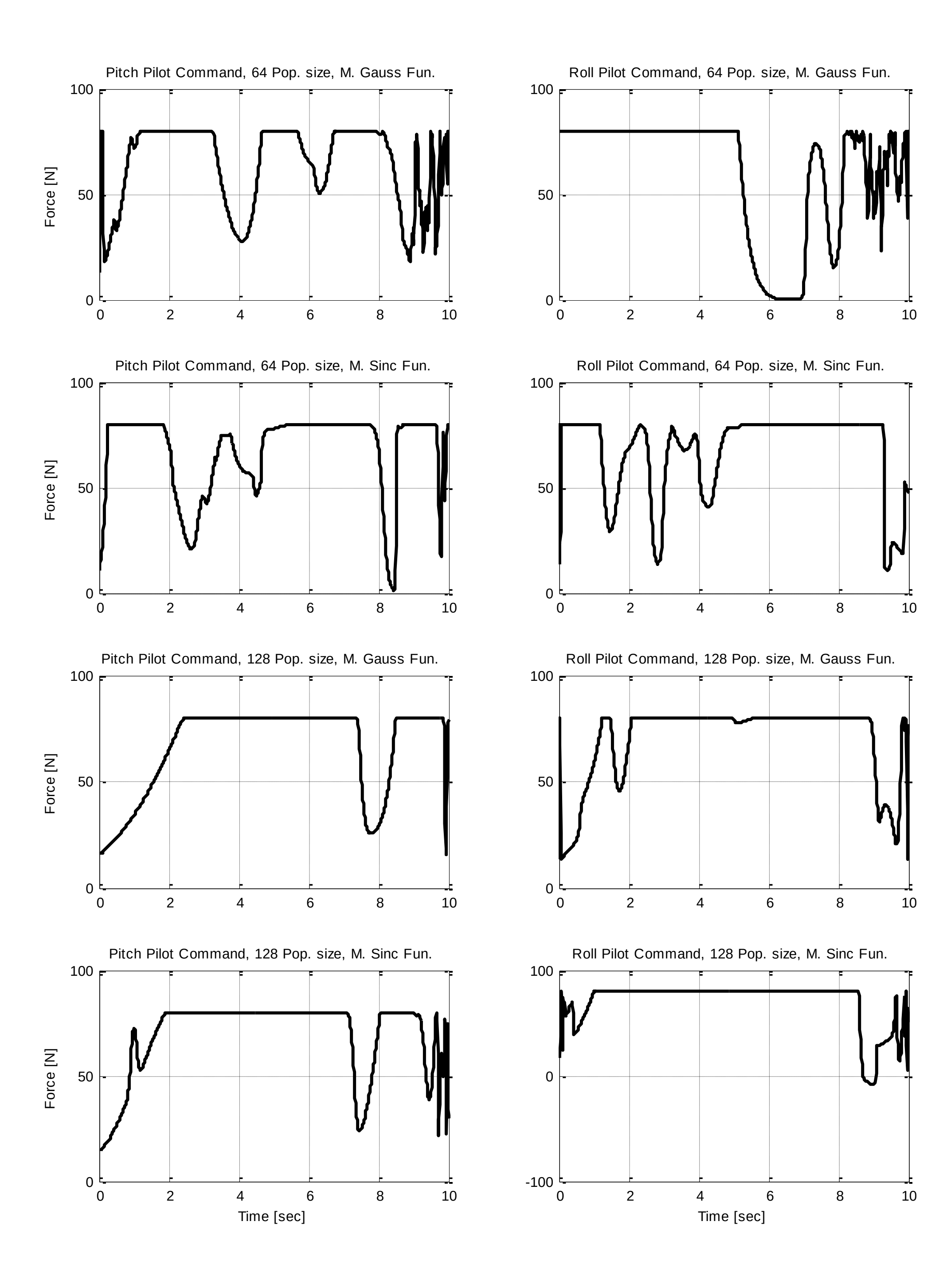


Figure 5.9: Pitch and Roll Pilot command obtained from CGA represent the pilot command that generate the maximum load factor (n_{zmax}) value. The aircraft model trimmed at 1000 m altitude and 0.3 mach speed.

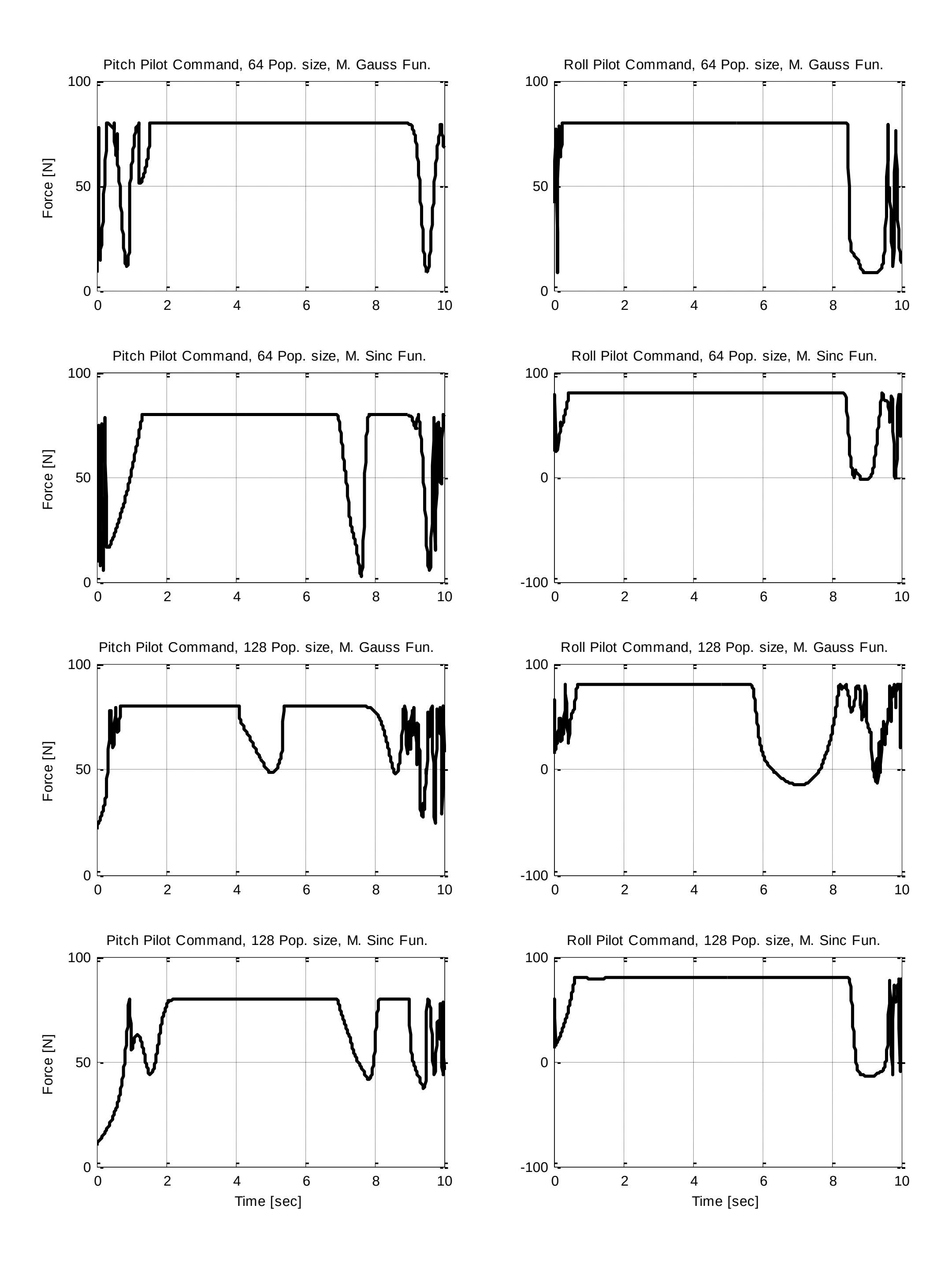


Figure 5.10: Pitch and roll pilot command obtained from CGA represent the pilot command that generate the maximum load factor (nzmax) value. The aircraft model trimmed at 6000 m altitude and 0.6 mach speed.

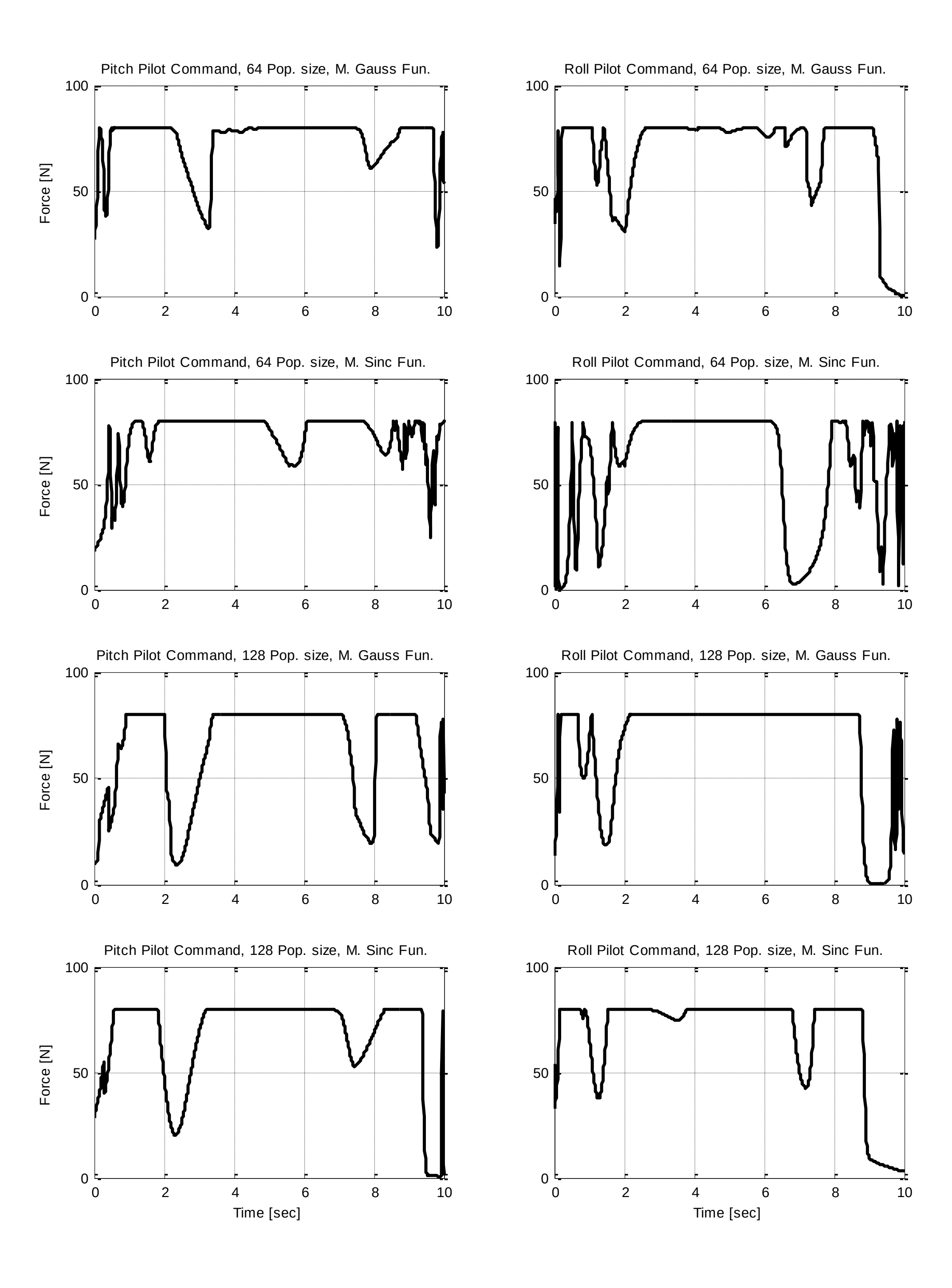


Figure 5.11: Pitch and roll pilot command obtained from CGA represent the pilot command that generate the maximum load factor (nzmax) value. The aircraft model trimmed at 3000 m altitude and 0.8 mach speed.

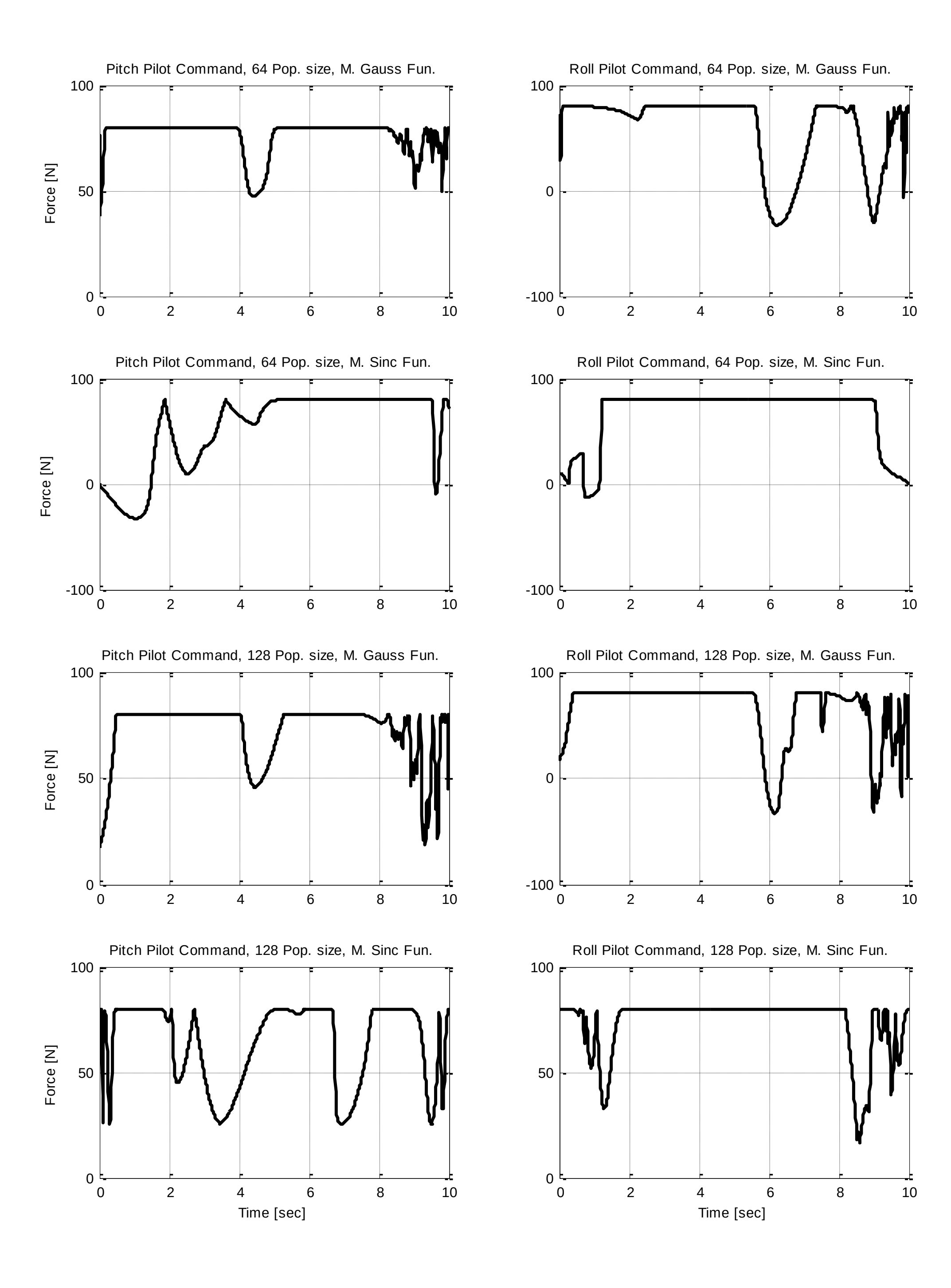


Figure 5.12: Pitch and roll pilot command obtained from CGA represent the pilot command that generate the maximum load factor (nzmax) value. The aircraft model trimmed at 5000 m altitude and 1.0 mach speed.

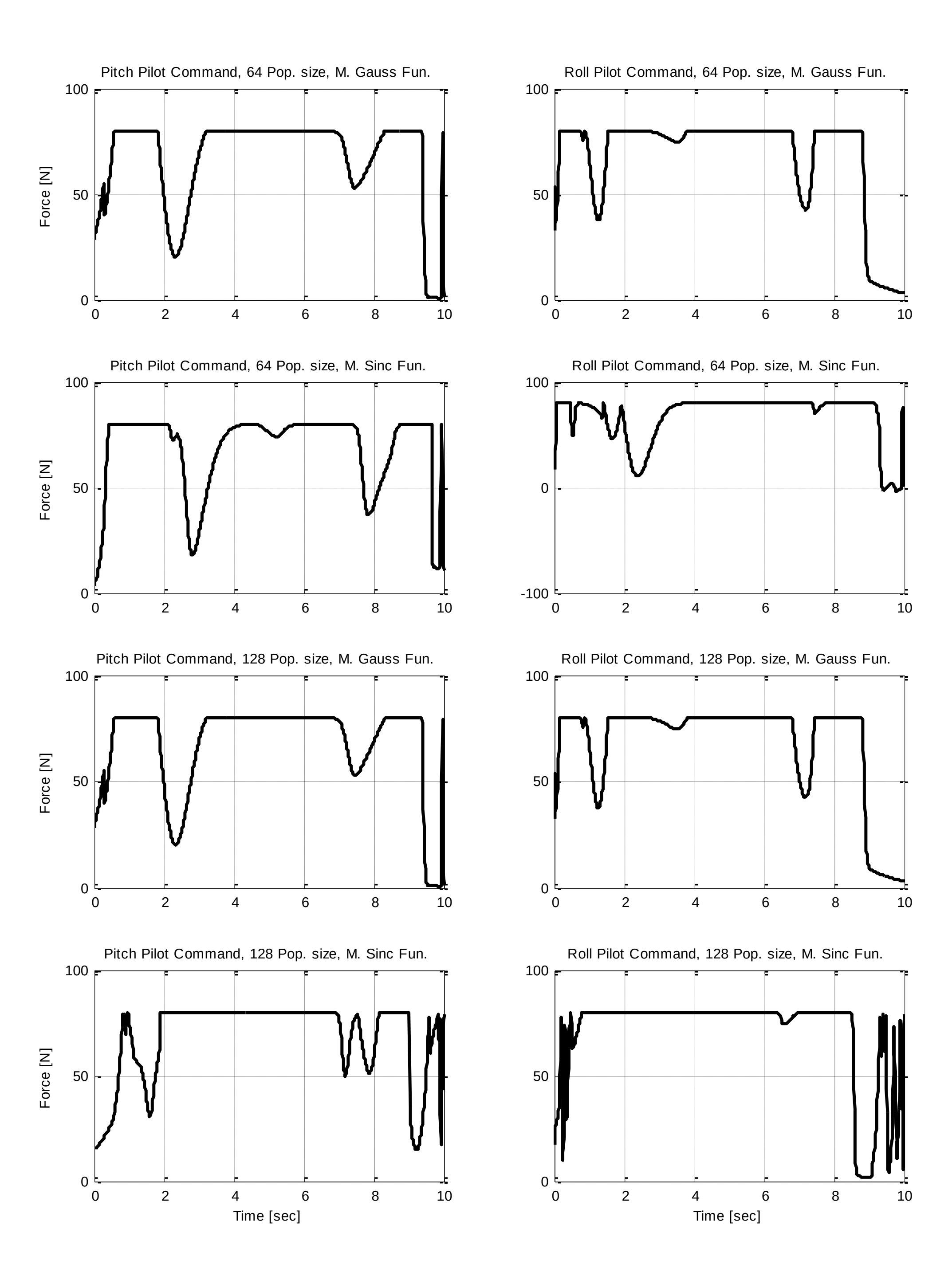


Figure 5.13: Pitch and roll pilot command obtained from CGA represent the pilot command that generate the maximum load factor (nzmax) value. The aircraft model trimmed at 3000 m altitude and 1.2 mach speed.

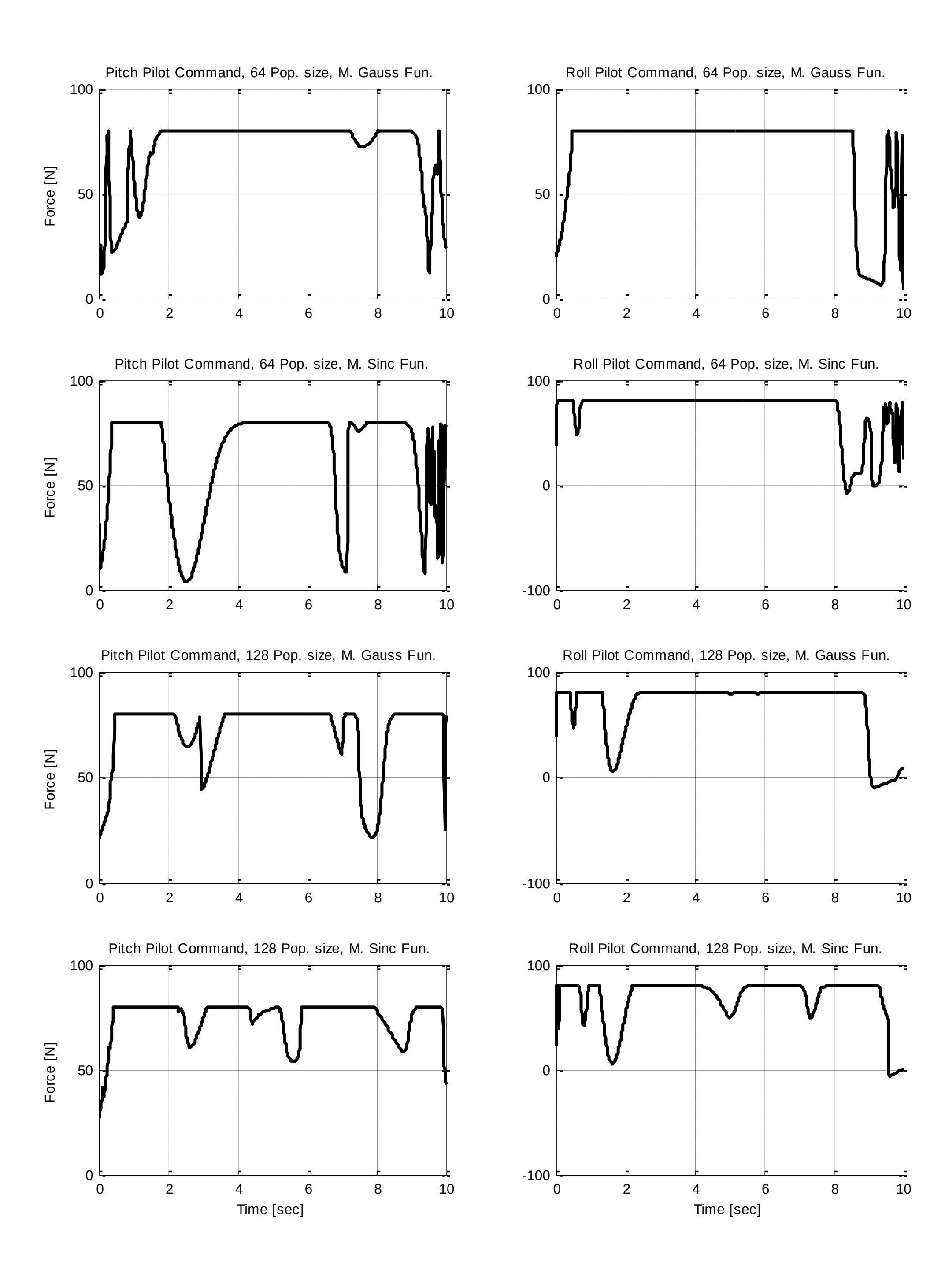


Figure 5.14: Pitch and roll pilot command obtained from CGA represent the pilot command that generate the maximum load factor (nzmax) value. The aircraft model trimmed at 6000 m altitude and 1.2 mach speed.

The above results show that the CGA has a significant convergence speed to the maximum value of the clearance criteria. For example, figure (5.15) shows the number of iterations (generations) in the above simulation results in figures (5.3-8) for the clearance criteria to reach 97.5% of the maximum value. As can be seen from this figure the majority of the simulation iteration reached 97.5% of the maximum value in about 20 iterations or less.

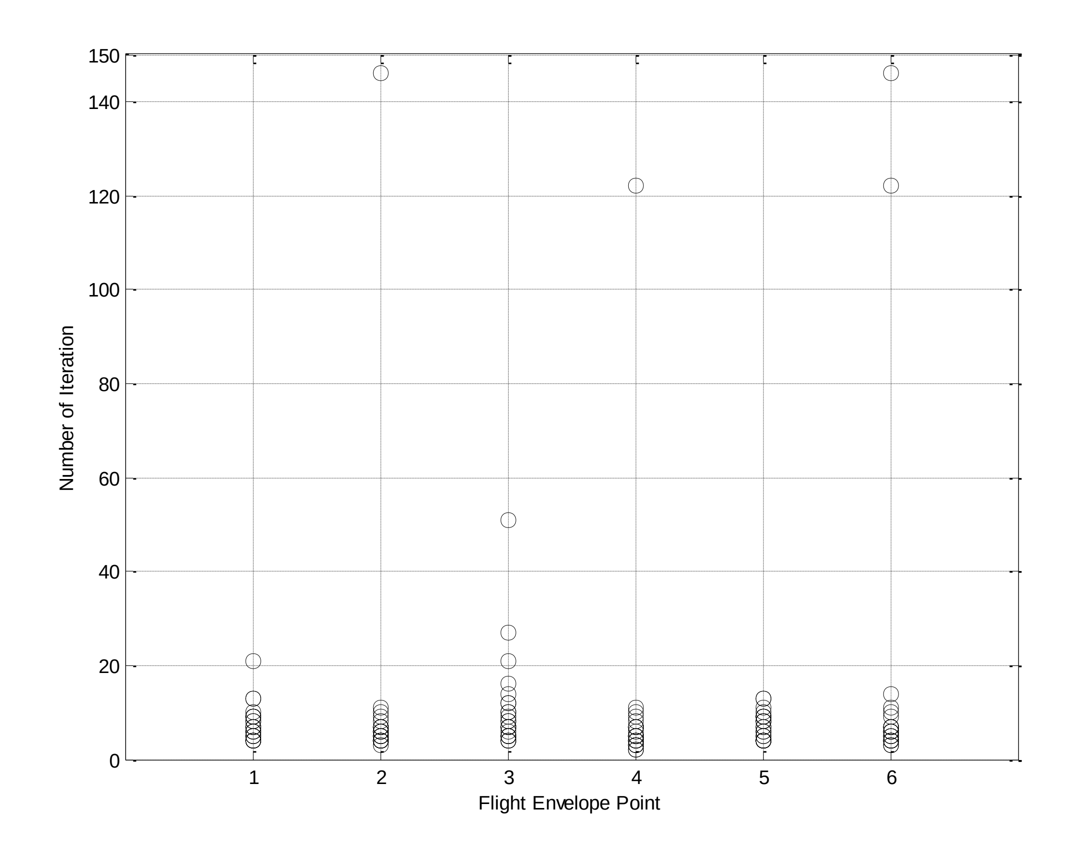


Figure 5.15. The convergence performance of CGA, flight envelope points corresponding number of iteration with load factor achieve 97.5% of maximum value of it.

5.5 Clearance of flight control laws based on Angle of Attack Criteria exceedance

The selected points in the flight envelope were processed and solved to find the maximum value for the Angle of Attack (A.O.A.), which can be obtained due to the pilot command inputs, the simulation carried out for only 64-population size.

Tables (6-7) and figures (5.16-27) show the simulation results when considering the angle of attack as fitness function in CGA. These results show that the maximum value for the angle of attack is 21.9 obtained at 6000 m and 1.2 Mach speed. This means the angle of attack criteria is within the limit (-10◦ - 26◦) specified by the clearance process [3]. This result agrees with that obtained by other investigators using other optimization methods.

In general the obtained results show that the flight control laws, which used in the aircraft dynamic model (ADMIRE) will protect the aircraft from the high values of angle of attack that results from the pilot command inputs

Table 5.6. The angle of attack simulation results for deferent runs using modified gauss function and the population size 64 individuals.

Mach	Alt	Run 1		Run 2		Run 3		Run 4		MAX Value
		Iter. No.	A.O.A	Iter. No.	A.O.A	Iter. No.	A.O.A	Iter. No.	A.O.A	
0.3	1000	208	20.39536	214	19.86801	171	19.35071	209	20.75794	20.75794
0.6	6000	140	19.39938	187	19.80721	148	18.83285	147	19.34686	19.80721
0.8	3000	221	20.04925	282	20.60796	308	21.66616	172	18.59303	21.66616
1	5000	179	20.00234	201	19.7998	150	19.12577	182	20.94341	20.94341
1.2	3000	276	17.59301	185	19.37977	162	20.19032	271	20.91565	20.91565
1.2	6000	212	21.90015	229	15.17864	208	20.39536	208	20.39536	**21.90015**

Table 5.7. The angle of attack simulation results for deferent runs using modified sinc function and the population size 64 individuals.

Mach	Alt	Run 1		Run 2		Run 3		Run 4		MAX Value
		Iter. No.	A.O.A	Iter. No.	A.O.A	Iter. No.	A.O.A	Iter. No.	A.O.A	
0.3	1000	182	19.30103	170	19.04308	251	20.60302	166	19.00672	20.60302
0.6	6000	210	20.24026	207	20.67776	262	20.89506	208	19.15628	20.89506
0.8	3000	175	13.64315	239	21.17564	210	20.24026	207	20.67776	**21.17564**
1	5000	210	20.24026	207	20.67776	262	20.89506	209	18.92149	20.89506
1.2	3000	200	21.0805	364	14.64127	192	19.00408	255	12.94459	21.0805
1.2	6000	450	20.90201	257	12.77573	178	19.91921	181	15.52381	20.90201

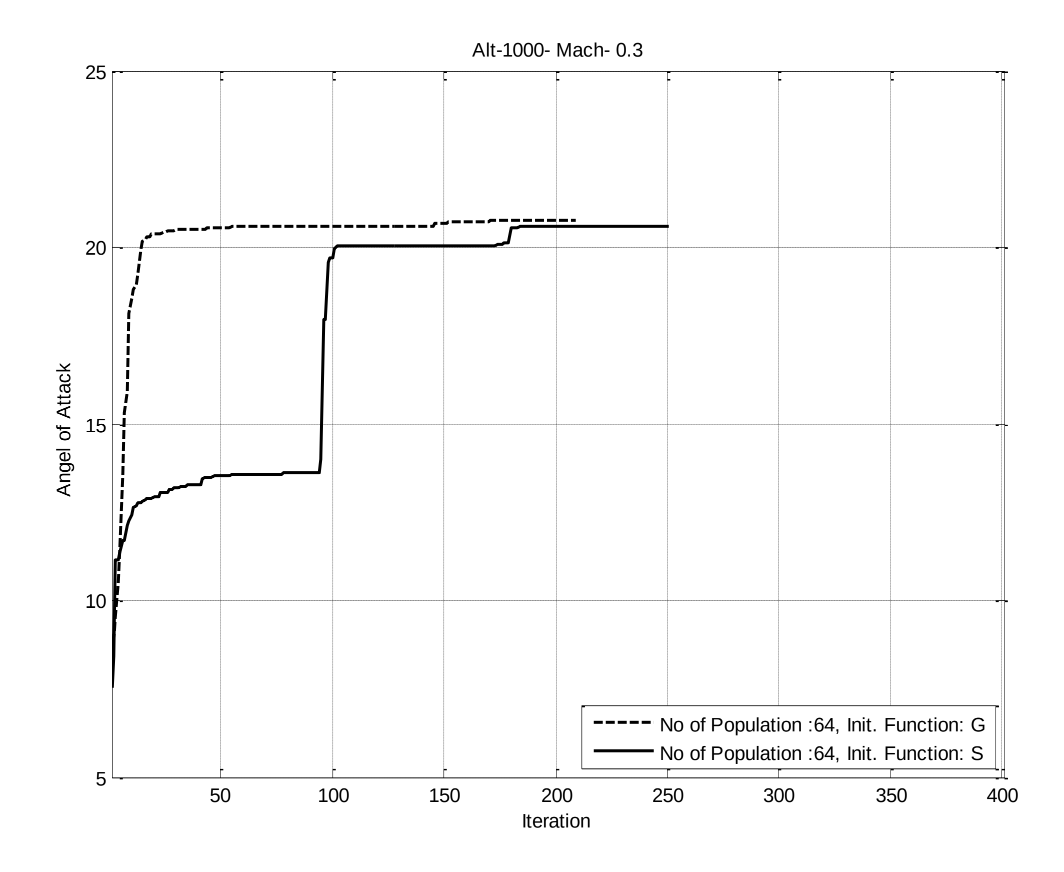


Figure 5.16: Maximum Angle of Attack versus the iteration (Generation). Result using CGA with population size, 64, and both initialization function, modified gauss function (G) and modified sinc function (S), aircraft model trimmed at 1000 m altitude and 0.3 mach speed.

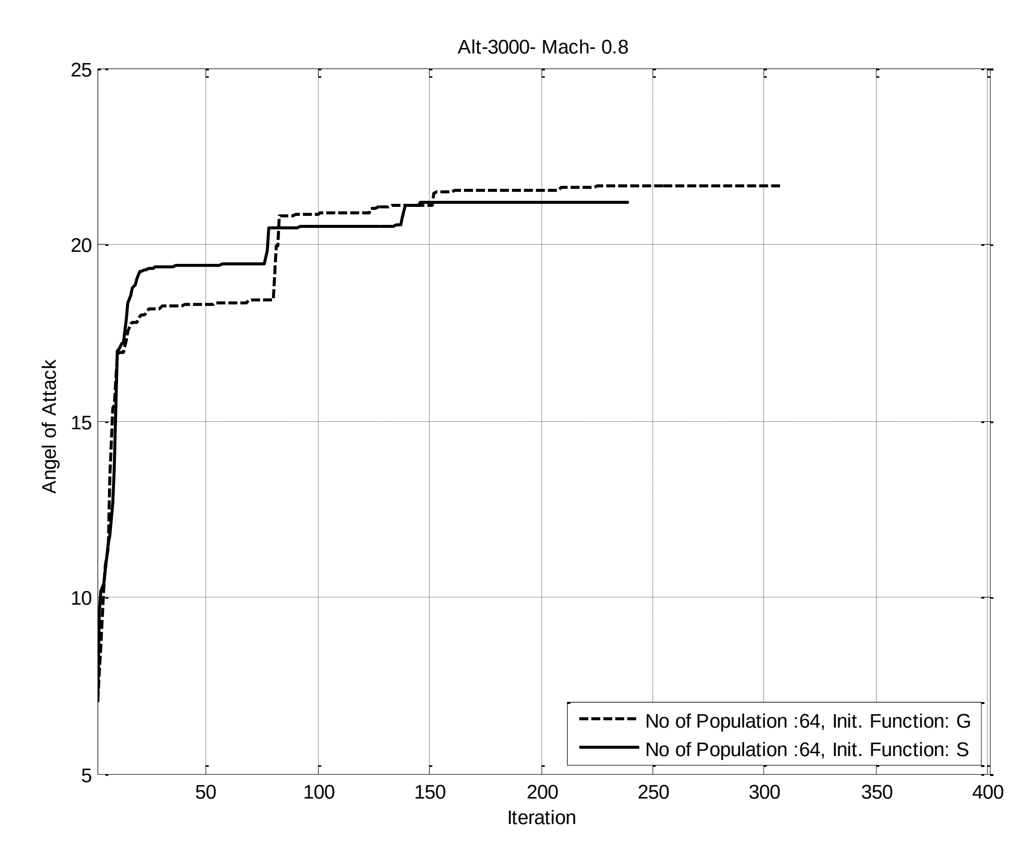


Figure 5.17: Maximum angle of attack versus the iteration (Generation). Result using CGA with population size, 64, and both initialization function, modified gauss function (G) and modified sinc function (S), aircraft model trimmed at 3000 m altitude and 0.8 mach speed.

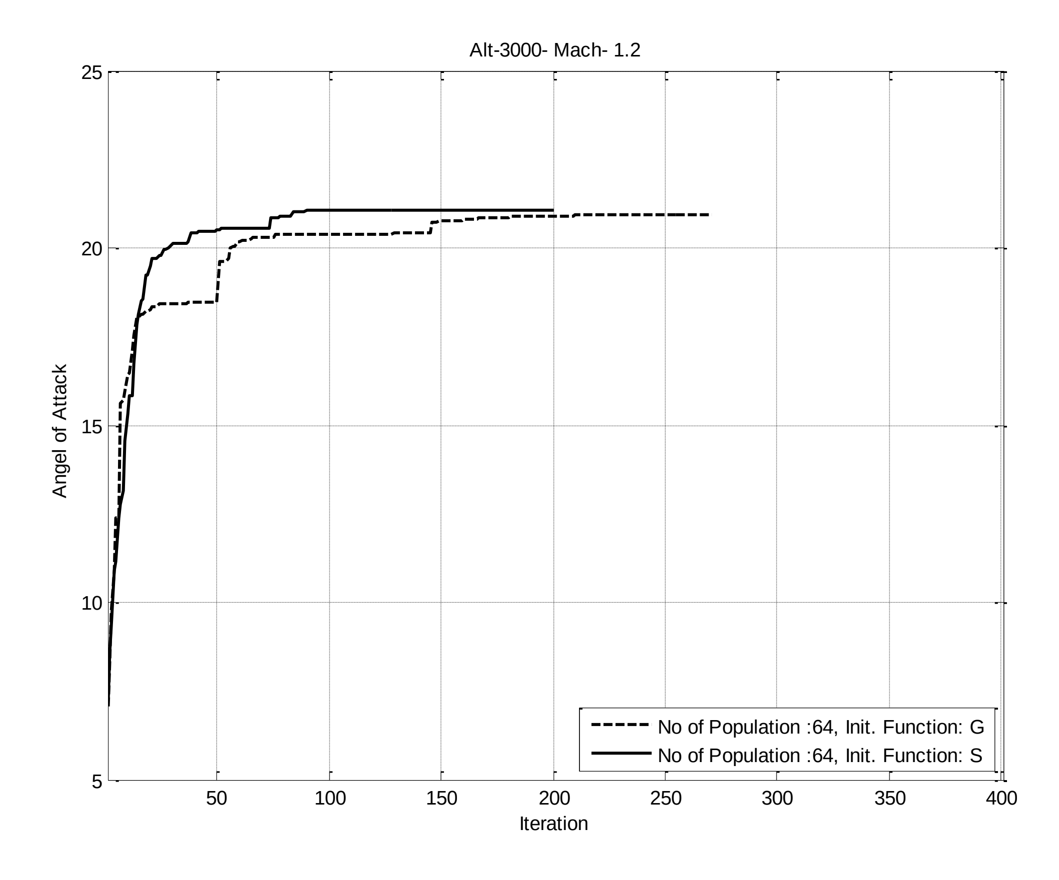


Figure 5.18: Maximum angle of attack versus the iteration (Generation). Result using CGA with population size, 64, and both initialization function, modified gauss function (G) and modified sinc function (S), aircraft model trimmed at 3000 m altitude and 1.2 mach speed.

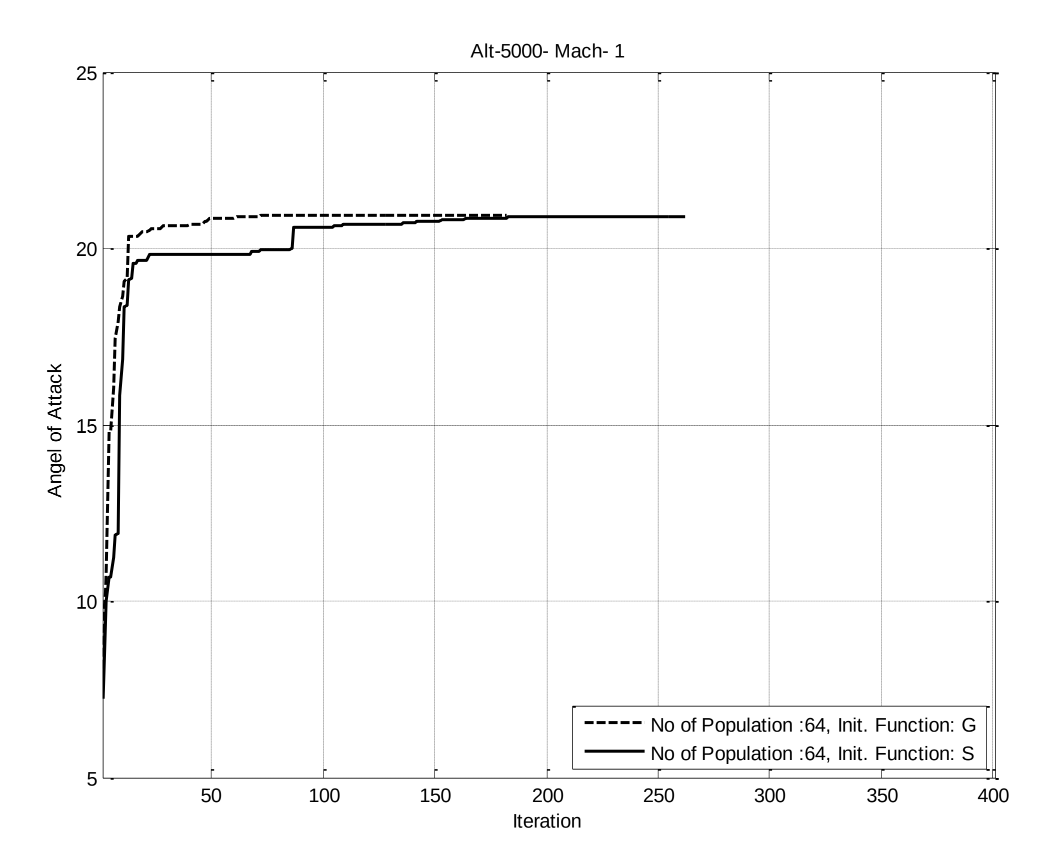


Figure 5.19: Maximum angle of attack versus the iteration (Generation). Result using CGA with population size, 64, and both initialization function, modified gauss function (G) and modified sinc function (S), aircraft model trimmed at 5000 m altitude and 1.0 mach speed.

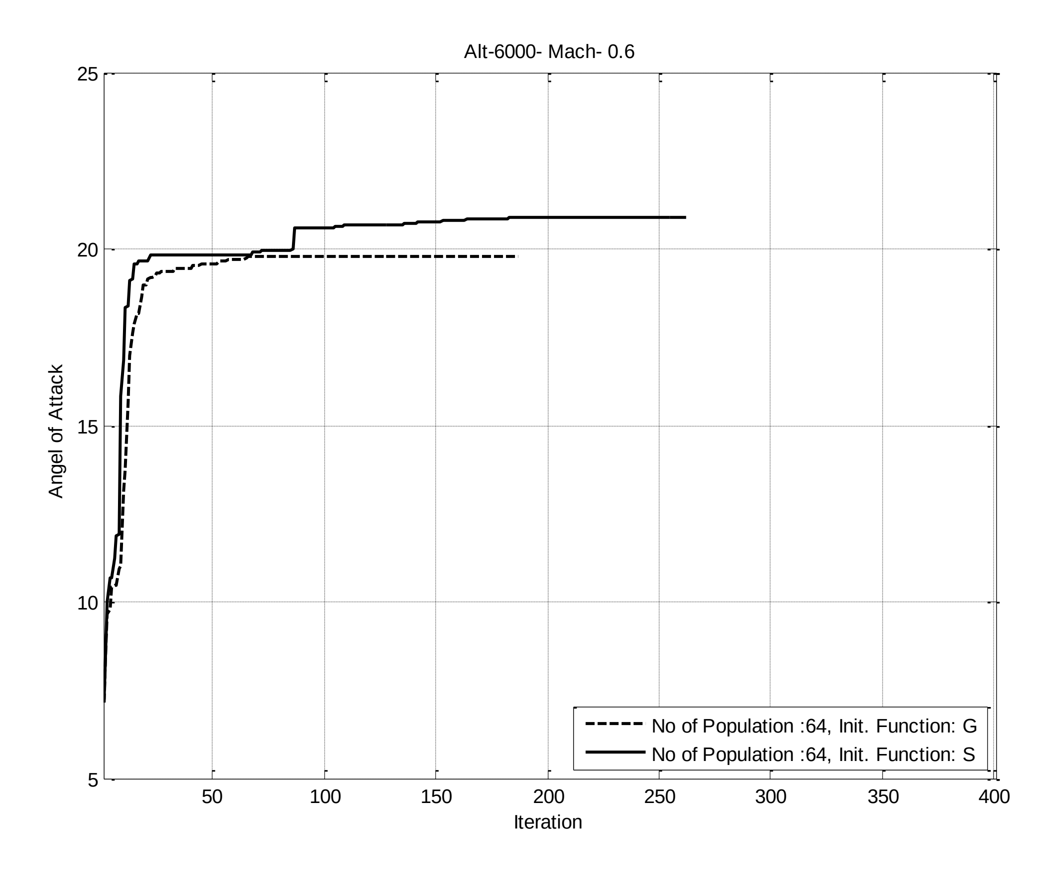


Figure 5.20: Maximum angle of attack versus the iteration (Generation). Result using CGA with population size, 64, and both initialization function, modified gauss function (G) and modified sinc function (S), aircraft model trimmed at 6000 m altitude and 0.6 mach speed.

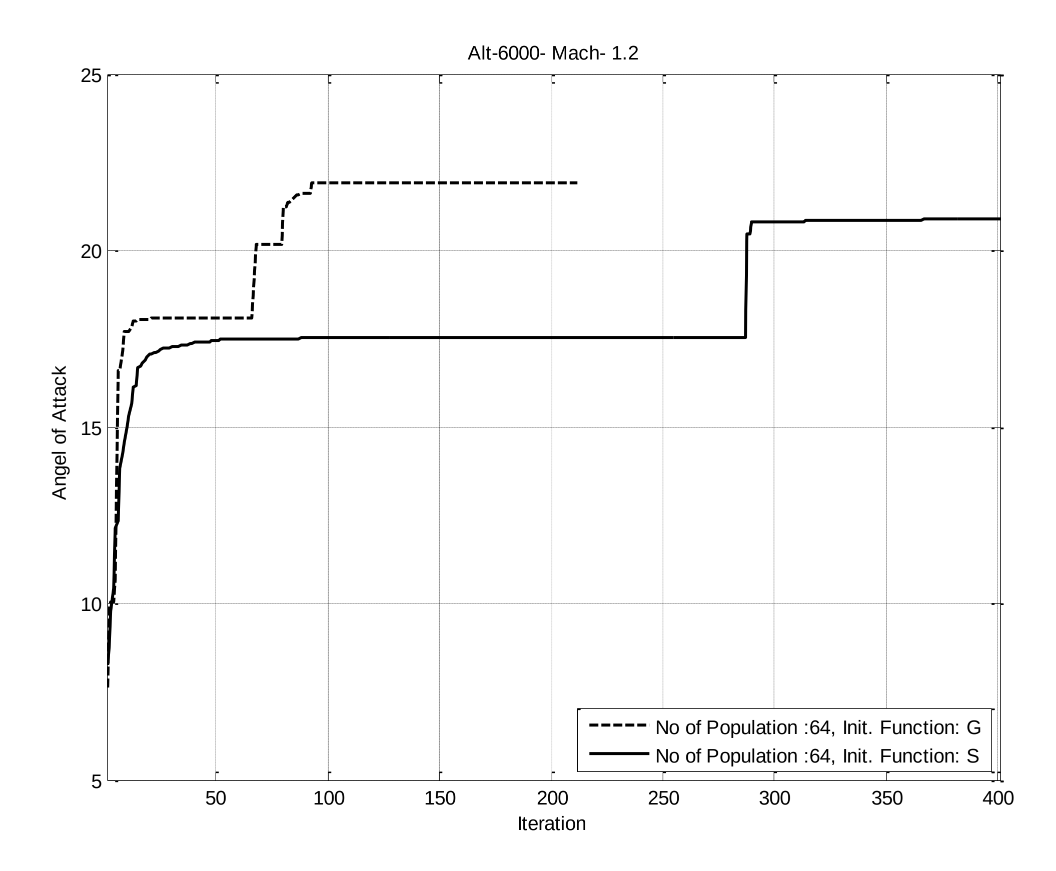


Figure 5.21: Maximum angle of attack versus the iteration (Generation). Result using CGA with population size, 64, and both initialization function, modified gauss function (G) and modified sinc function (S), aircraft model trimmed at 6000 m altitude and 1.2 mach speed.

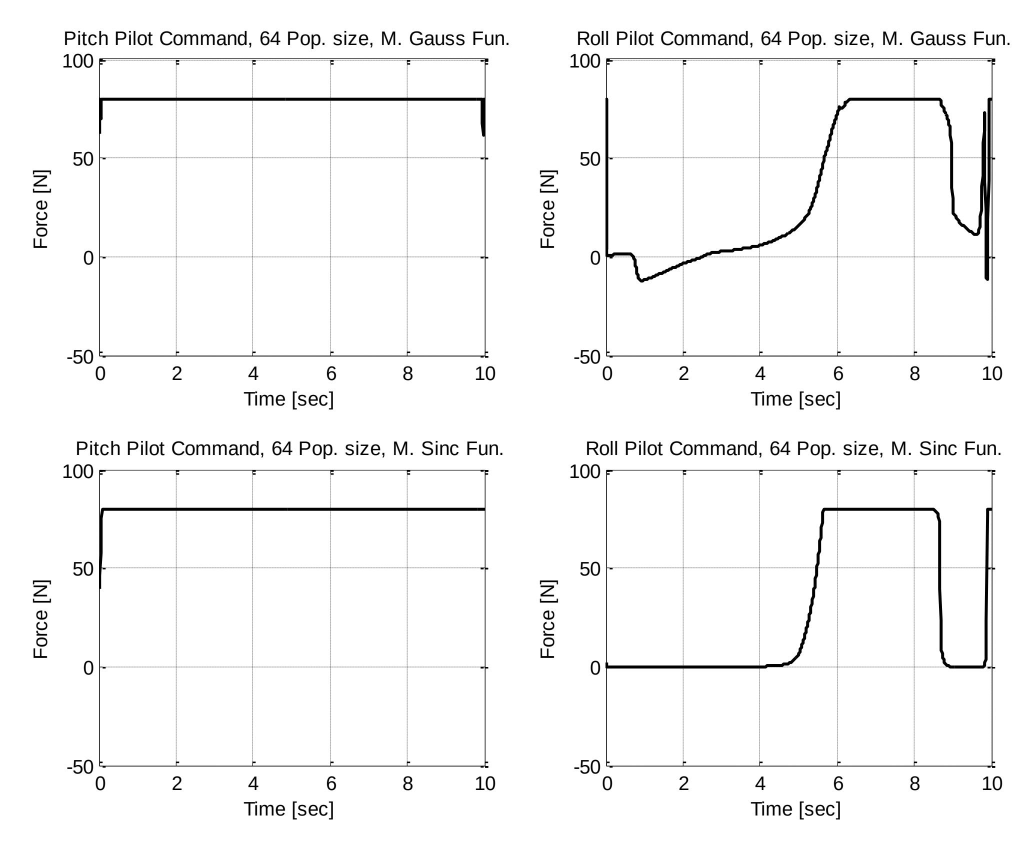


Figure 5.22: Pitch and roll pilot command obtained from CGA represent the pilot command that generate the maximum Angle of Attack value. The aircraft model trimmed at 1000 m altitude and 0.3 mach speed.

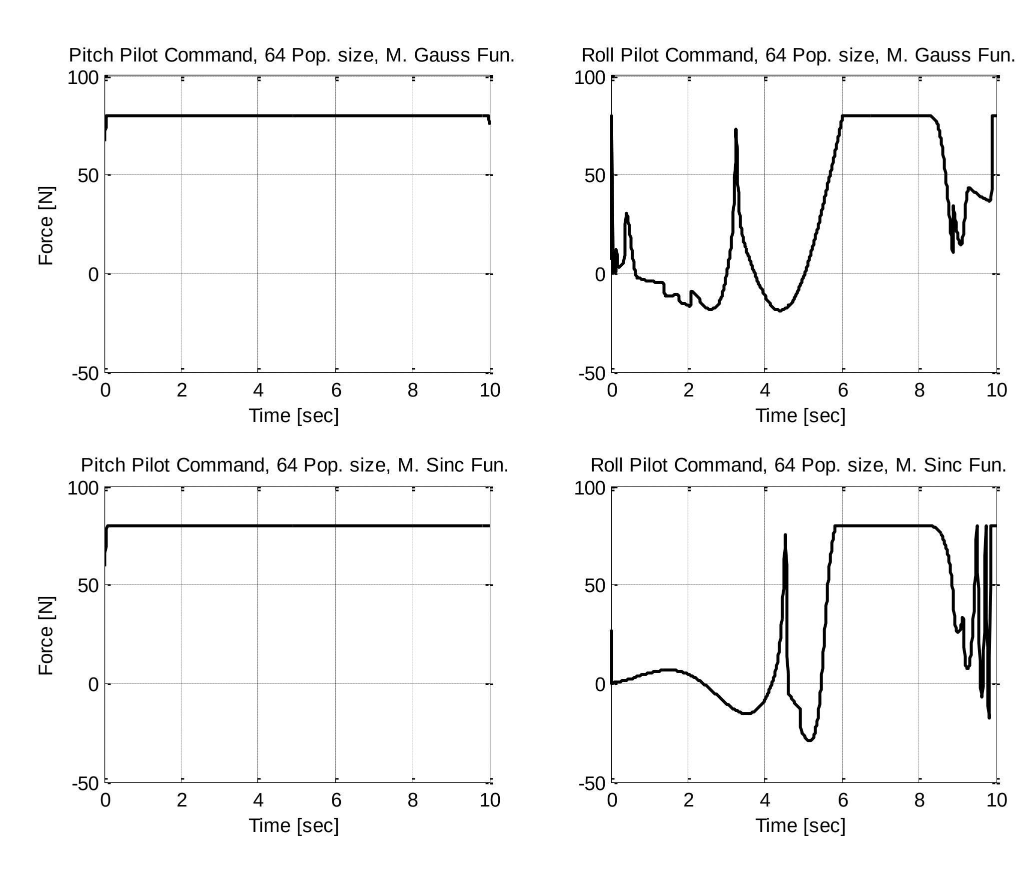


Figure 5.23: Pitch and roll pilot command obtained from CGA represent the pilot command that generate the maximum Angle of Attack value. The aircraft model trimmed at 6000 m altitude and 0.6 mach speed.

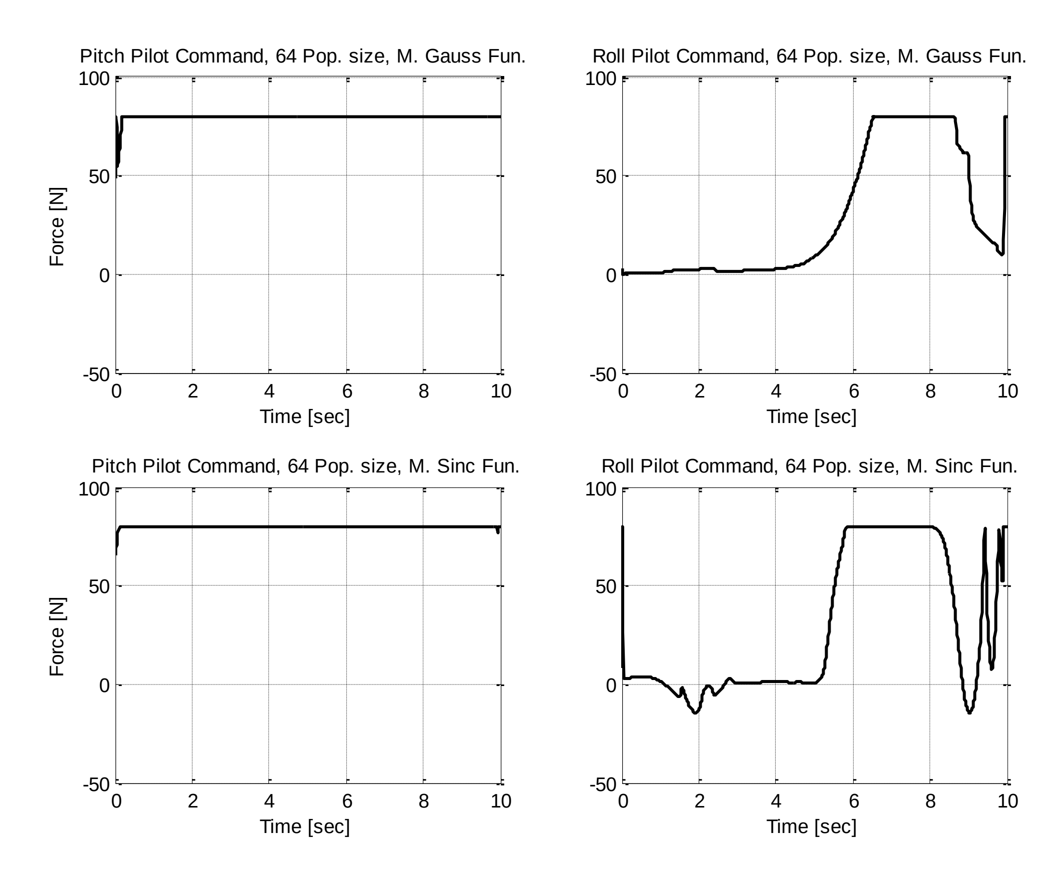


Figure 5.24: Pitch and roll pilot command obtained from CGA represent the pilot command that generate the maximum Angle of Attack value. The aircraft model trimmed at 3000 m altitude and 0.8 mach speed.

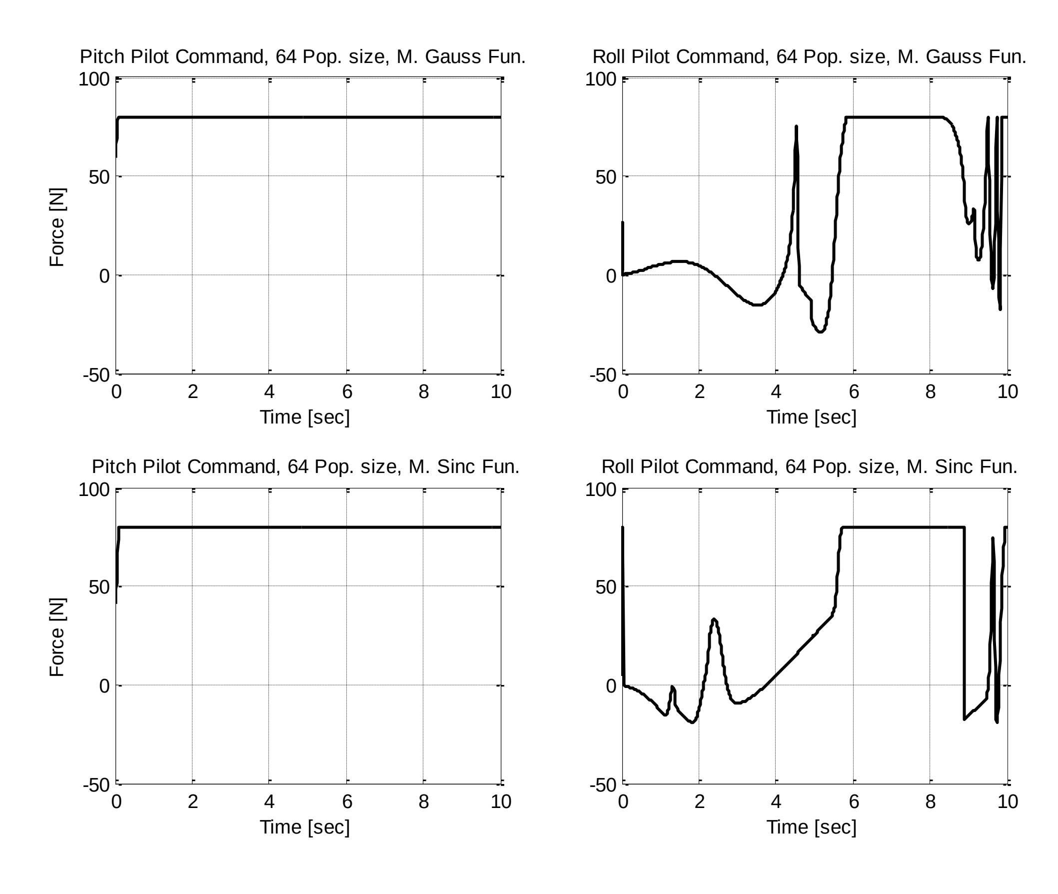


Figure 5.25: Pitch and roll pilot command obtained from CGA represent the pilot command that generate the maximum Angle of Attack value. The aircraft model trimmed at 5000 m altitude and 1.0 mach speed.

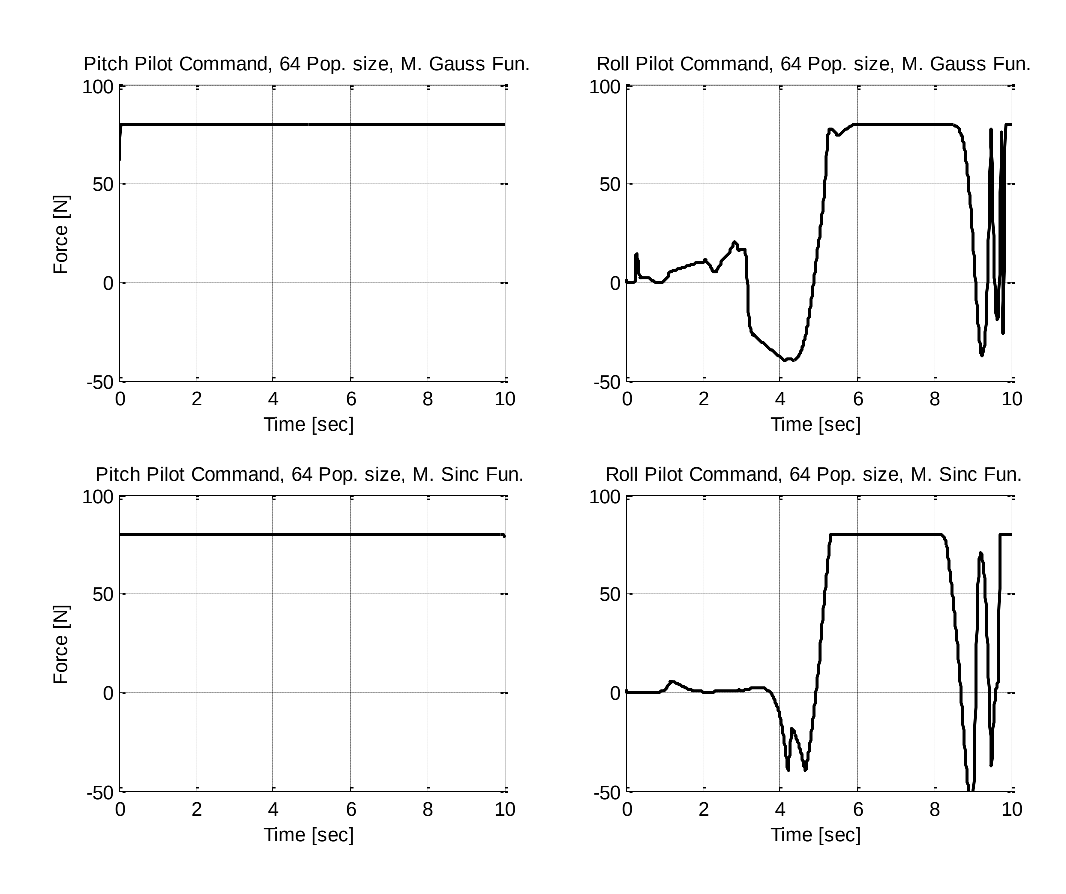


Figure 5.26: Pitch and roll pilot command obtained from CGA represent the pilot command that generate the maximum Angle of Attack value. The aircraft model trimmed at 3000 m altitude and 1.2 mach speed.

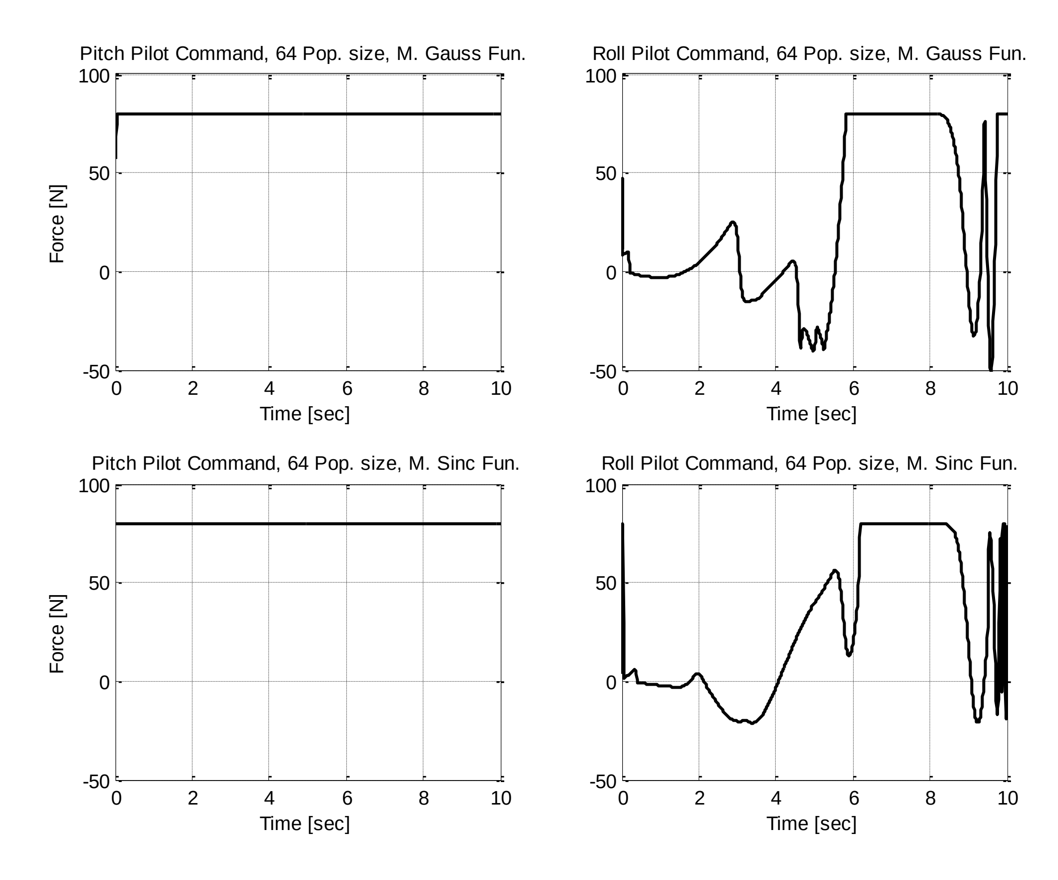


Figure 5.27: Pitch and roll pilot command obtained from CGA represent the pilot command that generate the maximum Angle of Attack value. The aircraft model trimmed at 6000 m altitude and 1.2 mach speed.

The above results show that the CGA has a significant convergence speed to the maximum value of the clearance criteria. For example, figure (5.28) shows the number of iterations (generations) in the above simulation results in figures (5.16-21) for the clearance criteria to reach 97.5% of the maximum value. As can be seen from this figure the majority of the simulation iteration reached 97.5% of the maximum value in about 100 iterations or less.

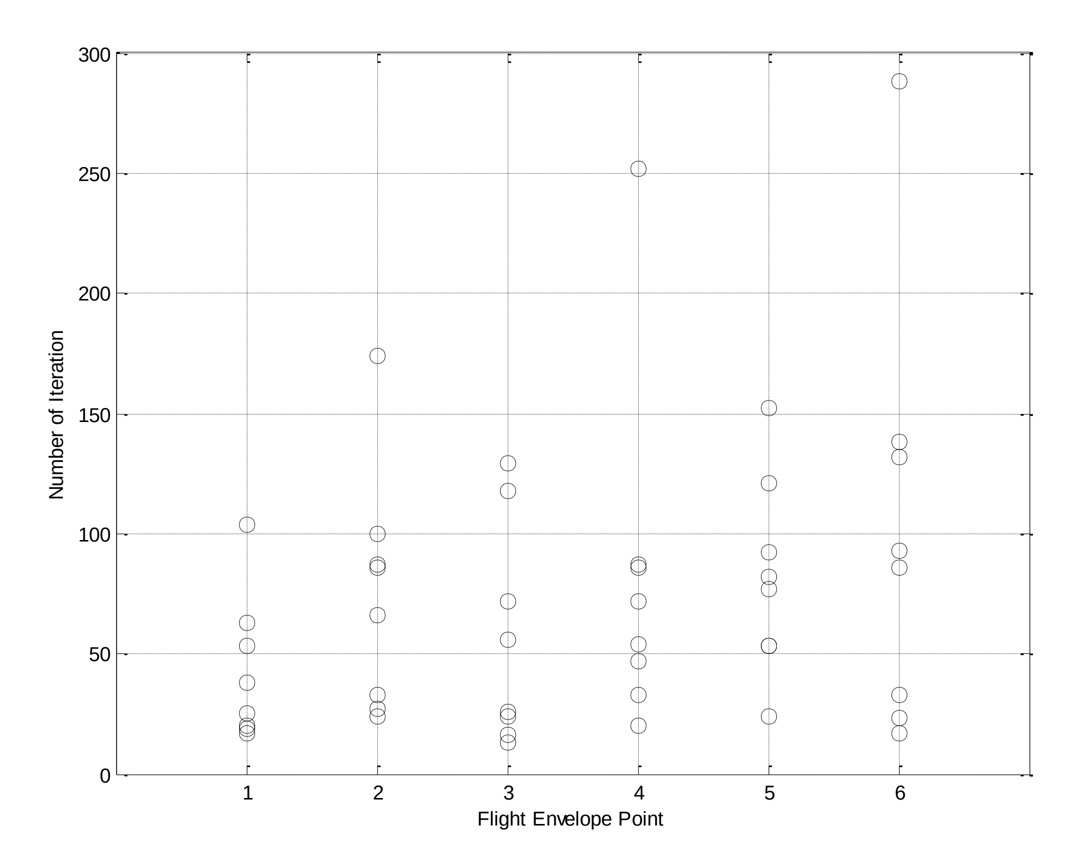


Figure 5.28. The convergence performance of CGA, flight envelope points corresponding number of iteration with angle of attack achieve 97.5% of maximum value of it.

5.6 Comparison between clearance process results based on CGA and GA

In literature, there are limited studied that deal with the pilot command input in the clearance process. The previous research, as discussed in chapter one, can be classified into the following: binary coded and real value coded representation of the pilot command input signal. The binary coded approach was used by Menon *et. al* [15], as it shown in figure (5.29) the pilot command signal is discretized into four level and each level has its binary representation. Menon used this approach to solve the optimization problem that result from the clearance process based on the pilot command input and the parameter variation in the uncertainties vector.

The second class, the real coded representation, was used by Skoog *et. al* [24], as it is shown in figure (5.30), they use the linear parameterization to represent the pilot command input, then solved the optimization problem using the GA. The maximum value for the load factor that obtained by Skoogh is 10.89g.

In comparison to the results that obtained using CGA, the CGA has results values higher than those obtained using GA by Skoogh. In addition, CGA has significant convergence speed to the maximum value of the clearance criteria relatively to that presented in both approaches.

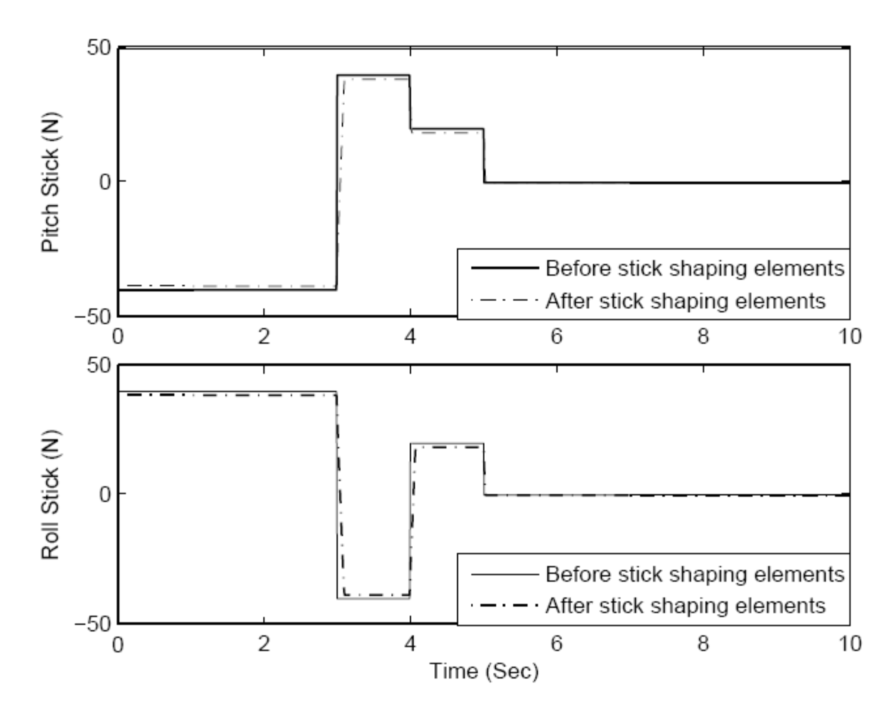


Figure 5.29: Pilot command signal representation that used by Menon.

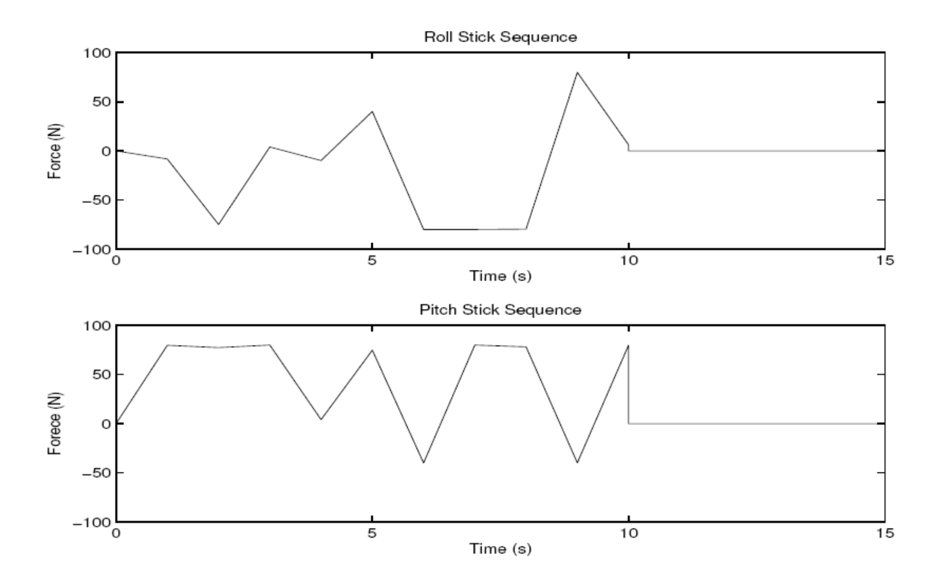


Figure 5.30: Pilot command signal representation that used by Skoogh.

.

5.7 Parallel Programming in CGA

In order to reduce the computation time during running of the simulation, parallel programming was used in the CGA. In the parallel programming, the computation tasks were distributed to the available central processing unit (CPU).

The simulation was performed using Laptop equipped with Intel Core i3 CPU, M330@2.13 GHz and 4.0GB Random access memory (RAM), this system give the option to have 2,3, and 4 CPU.

Figure (5.31) shows the computation time with respect to the population size using 2, 3, and 4 CPU in addition to the normal setting with one CPU. By using the parallel programming, the computation time was reduced by more than 50% of the normal programming. Regarding the number of population, the percentage is fixed.

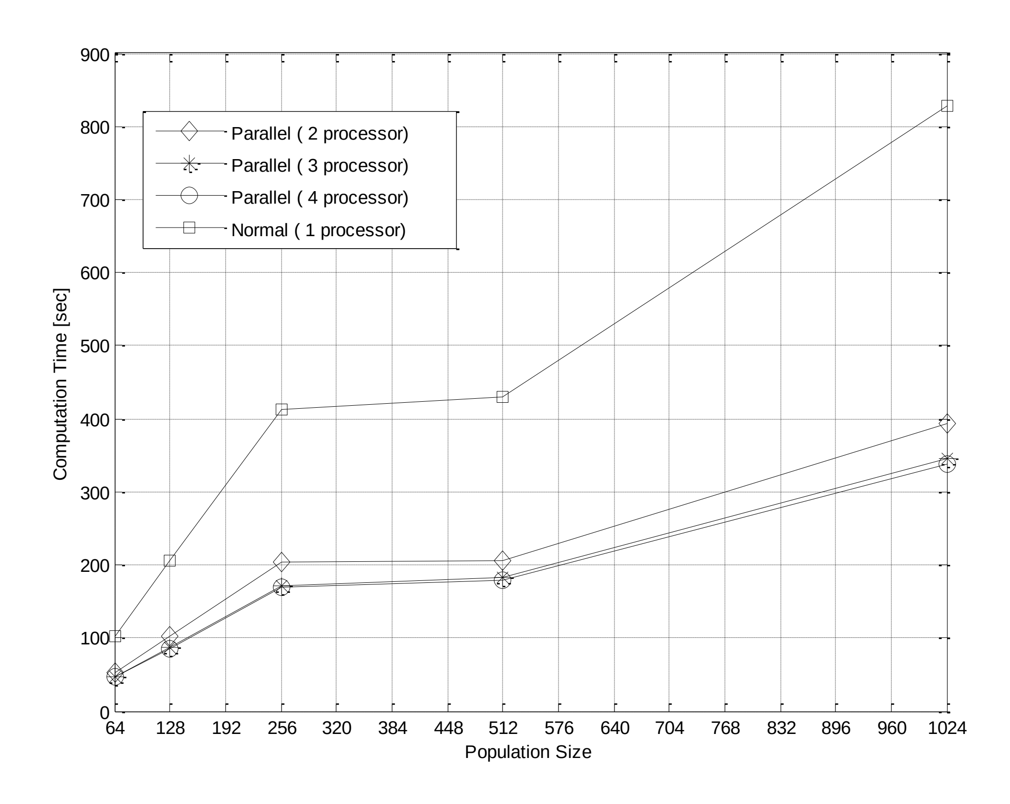


Figure 5.31: The computation time for each iteration with respect to the population size.

5.8 Chapter Summary

In this chapter, the results that obtained from the MATLAB simulation presented. The first part of the simulation results shows the results for flight control laws clearance based on load factor criteria, the simulations for this case were carried out for all flight envelope with 64 and 128 individuals population size. The results shows that the flight control laws fail, (not clear), to control the aircraft response within the safe limit of load factor.

The second part of the simulation shows the results for the flight control laws based on the angle of attack criteria. In this simulation, the population size was selected 64 individuals only and the stopping criteria selected 500 maximum generation. The results prove that the flight control laws which was augmented in ADMIRE are clear and can handle the aircraft in safe operation.

The implementation of parallel programming and its effect in computation time in CGA were discussed.

CHAPTER SIX

CONCLUSION AND FUTURE WORK

6.1 Conclusion

In this work, CGA is used as global optimization problem solver to study the clearance process of flight control law. The CGA is used to find the worst-case pilot inputs found based on the load factor and angle of attack exceedance criteria, for a given speed and altitude that might stall the aircraft or exceed its structural stress limit. The results presented in this work agreed with those obtained by other investigators using other optimization methods. In addition the results in this investigation shows the CGA has a significant convergence speed to the maximum value of the considered clearance criteria for angle of attack and load factor.

Several research use conventional Genetic algorithms and other optimization methods to find the worst-case combination but with some restriction on the pilot command signal such as parameterize the pilot signals and then solve the resulting optimization problem. However, here in CGA there is no any restriction on the shape of the pilot command signal.

6.2 Future Work

The future works have two interconnected phases. The first phase related to the clearance process itself. In this, phase the uncertainty vector to be taken into account during clearance of the pilot command signal. This phase needs to reconfigure the algorithms to deal with such type of parameters.

The second phase related to the CGA algorithm. Improve the selection schemes and the other related CGA operators in order to increase the efficiency and the convergence speed of CGA.

REFERENCES

[1] Abo-Hammour Z. S., M. Yusuf, N. M. Mirza, S. M. Mirza, M. Arif, (2002). **Cartesian path generation of robot manipulators using continuous genetic algorithms**, Robotics and Autonomous Systems 41(40), pp 179-223.

[2] Abo-Hammour Z. S., M. Yusuf, N. M. Mirza, S. M. Mirza, M. Arif, (2004). **Numerical solution of second-order, two-point boundary value problems using continuous genetic algorithms**, Int. J. Numer. Meth. Eng. 61, pp 1219-1242.

[3] Abo-Hammour, Z. S.; ASASFEH, A. GH.; Al-Smadi, A. ; Alsmadi, O.,(2010), **A Novel Continuous Genetic Algorithm for the Solution of Optimal Control Problems**, Optimal Control, Applications and Methods, John Wiley.

[4] Chetty, S., Deodhare G. and Misra B. B. (2002). **Design, development and flight testing of control laws for the Indian Light Combat Aircraft**. In: Proc. AIAA Atmospheric FM Conference. CP-3810, CA, USA.

[5] Davidor Y., (1991). **Genetic algorithms and robotics: a heuristic strategy for optimization**, World Scientific.

[6] de Pinho M. do R. , Vinter R. B., (1997). **Necessary conditions for optimal control problems involving nonlinear differential algebraic equations,** Journal of Mathematical, Analysis, and Applications 212, pp. 493-516.

[7] Debasis S., Jayant M. Modak, (2004). **Genetic Algorithms with Filters for Optimal Control Problems in Fed-batch Bioreactors**, Bioprocess Biosyst. Eng, 26, pp. 295-306.

[8] Fielding C., Varga A., Bennani S., and Selier M. (Eds.), (2002), **Advanced techniques for clearance of flight control laws**, Springer, 2002.

[9] Forssell L. and Nilsson U., (2005), **ADMIRE The Aero-Data Model In a Research Environment Version 4.0, Model Description**, FOI – Swedish Defense Research Agency Systems Technology SE-164 90 STOCKHOLM.

[10] Forssell L. and Hyden A., (2003), **Flight control system validation using global nonlinear optimization algorithms,** Proc. of the European Control Conference, Cambridge, U.K., September 2003.

[11] Goldberg D. E, (1989**). Genetic algorithms in search, optimization, and machine learning,** Addison-Wesley.

[12] Gutowski M.W., (1994), **Smooth Genetic Algorithms,** Journal of Physics, Mathematical and general, 27, 7893-7904.

[13] Hagstrom Martin, (2007). **The ADMIRE Benchmark Aircraft Model**, Nonlin. Anal. & Syn. Tech. for Aircraft Ctrl., LNCIS 365, pp. 35–54.(springerlink.com).

[14] Karlsson Fredrik, (2002), **Flight Nonlinear Flight Control Design and Analysis Challenge,** Control System Department, Saab AB, SE-581 88 Linkoping, Sweden.

[15] Menon P.P., Kim J., Bates D.G., and Postlethwaite I., (2005), **Computation of worst-case pilot inputs for clearance of flight control laws**, 16th Triennial World Congress, Prague, Czech Republic.

[16] Menon P.P., Kim J., Bates D.G., and Postlethwaite I., (2006). "**Clearance of nonlinear flight control laws using hybrid evolutionary optimization**". IEEE Transactions on Evolutionary Computation, 10(6):689–699.

[17] Menon. P.P., Kim J., Bates D.G., and Postlethwaite I., (2004). **"Improved Clearance of Flight Control Laws Using Hybrid Optimization",** Proc. of the IEEE Conference on Cybernetics and Intelligent Systems, Singapore, December, 2004.

[18] Montero G., (1997). **Solving Optimal Control Problems by GAs,** Nonlinear Analysis, theory, Methods & Applications, Vol,30 No 5, pp. 2891-2902.

[19] Oosterom M., Babuska R., (2006), **Design of a gain-scheduling mechanism for flight control laws by fuzzy clustering**, Control Engineering Practice 14,pp. 769–781.

[20] Pratt R. W., (2000), **FLIGHT CONTROL SYSTEMS practical issues in design and implementation**, The American Institute of Aeronautics and Astronautics.

[21] Ryan G. W. III, (1995), **A Genetic Search Technique for Identification of Aircraft, Departures,** Dryden Flight Research Center Edwards, California Under Contract NAS 2-13445.

[22] Selier M., Fielding C. (BAE Systems), U. Korte (EADS), R. Luckner (Airbus), (2003). **New Analysis Techniques for Clearance of Flight Control Laws**, AIAA Guidance, Navigation and Control Conference, Austin, Texas, U.S.A., 11-14 August, 2003.

[23] Seywald H. , R. R. Kumar, S. M. Deshpande, (1995). **Genetic Algorithm approach for optimal control problems with linearity appearing controls**, Journal of Guidance, Control, and Dynamics, Vol. 18, No. 1, pp. 177-182.

[24] Skoogh D., Eliasson P., Berefelt F., R. Amiree, D. Tourde, Forssell L. (2009) **“Clearance of flight control laws for time varying pilot input signals”**. Proceedings of the 6th IFAC Symposium on Robust Control Design, Haifa, Israel, June 16-18.

[25] Stevens B. L. and Lewis F. L., **Aircraft Control and Simulation**. John Wiley & Sons, Inc., New York, 2nd edition, 1992.

[26] Vooren A. (2003) **Expanding ADMIRE’s Aerodynamic Envelope for High Angles of Attack**. Technical Report FOI-R--0771--SE, Swedish Defense Research Agency, Stockholm.

[27] Yamashita Y. , Shima M., (1997). **Numerical computational method using genetic algorithm for the optimal control problem with terminal constraints and free parameters**, Nonlinear analysis theory, Methods & Applications, Vol 30, No 4, pp. 2285-2290.

[28] Yechout R., Morris L., Bossert E., and Hallgren F. (2008), **Introduction to Aircraft Flight Mechanics: Performance, Static Stability, Dynamic Stability, and Classical Feedback Control,** American Institute of Aeronautics and Astronautics, Inc.

[29] Zbigniew M., Cezary Z.,(1992). **A Modified Genetic Algorithm For Optimal Control Problems,** computers Math, Applic, Vol 23, No, 12, pp. 83-94.

[30] Zbigniew M., Krawczyk B. , Kazemi M., Cezary Z., (1990). **Genetic Algorithms and optimal control problems**, proceeding of the 29th IEEE Conference on Decision and Control, 1-6, pp. 1664-1666.

APPENDIX A. STATISTICAL CONTROL ANALYSIS

Statistical analysis was performed for each point in the flight envelope, which were considered in chapter 5. This analysis is based on Xbar charts in order to show that all the runs result were inside the control box of these charts. The results of this analysis are shown in tables A1-6 and figures B1-6 for deferent points in the flight envelope considered in chapter 5. ULC is the upper control box limit corresponding to three standard deviations from the mean and LCL is the lower control box limit corresponding to minus three standard deviations from the mean. As can be seen from these results all points lie inside the statistical control box, which confirms that the present CGA results agree with the standard distribution curve.

Table A1. Load factor for flight envelope point #1: 0.3 Mach & 1000 m Altitude.

Initialization Function	Population Size	Run1	Run2	Run3	Run4	**Mean**	**Standard Deviation**	**UCL**	**LCL**
		Nz	Nz	nz	nz				
M. Gauss Function	64	10.558	10.42	10.569	10.426	**10.570**	**0.1531**	**11.030**	**10.111**
M. Sinc Function	64	10.65	10.828	10.612	10.562				
M. Gauss Function	128	10.508	10.268	10.436	10.469				
M. Sinc Function	128	10.636	10.583	10.813	10.786				

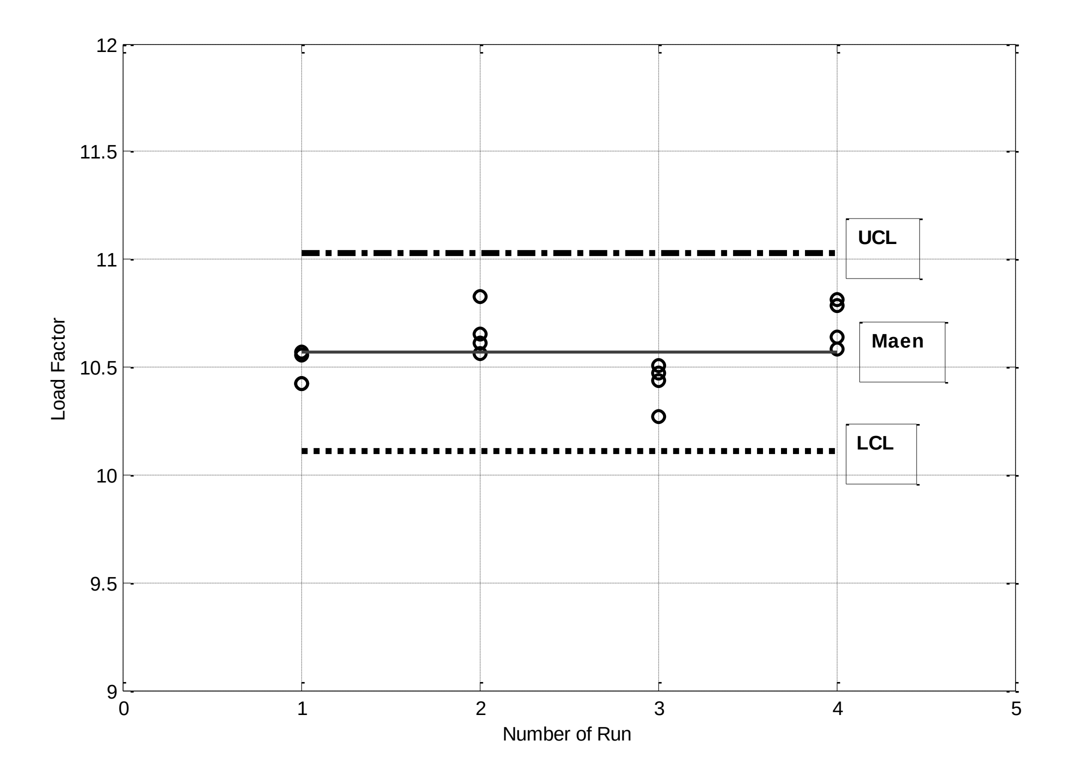


Figure A1: Xbar chart for point #1.

Table A2. Load factor for flight envelope point #1: 0.6 Mach & 6000 m Altitude.

Initialization Function	Population Size	Run1	Run2	Run3	Run4	**Mean**	**Standard Deviation**	**UCL**	**LCL**
		Nz	nz	nz	nz				
M. Gauss Function	64	10.404	10.443	10.473	10.42	**10.557**	**0.151**	**11.011**	**10.104**
M. Sinc Function	64	10.48	10.721	10.812	10.517				
M. Gauss Function	128	10.47	10.496	10.348	10.661				
M. Sinc Function	128	10.595	10.73	10.842	10.503				

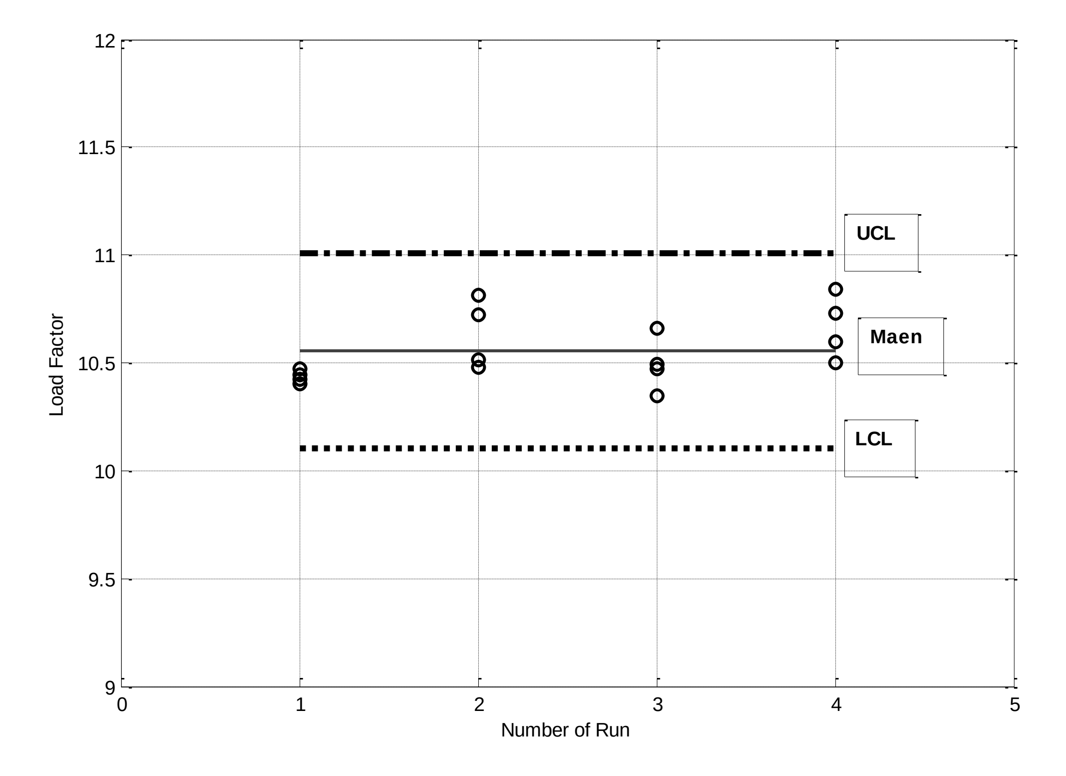


Figure A2: Xbar chart for point #2.

Table A3. Load factor for flight envelope point #1: 0.8 Mach & 3000 m Altitude.

Initialization Function	Population Size	Run1	Run2	Run3	Run4	**Mean**	**Standard Deviation**	**UCL**	**LCL**
		Nz	nz	nz	nz				
M. Gauss Function	64	10.642	10.399	10.712	10.618	**10.59**	**0.163**	**11.08**	**10.1**
M. Sinc Function	64	10.6	10.569	10.408	10.535				
M. Gauss Function	128	10.829	10.653	10.809	10.839				
M. Sinc Function	128	10.425	10.449	10.664	10.301				

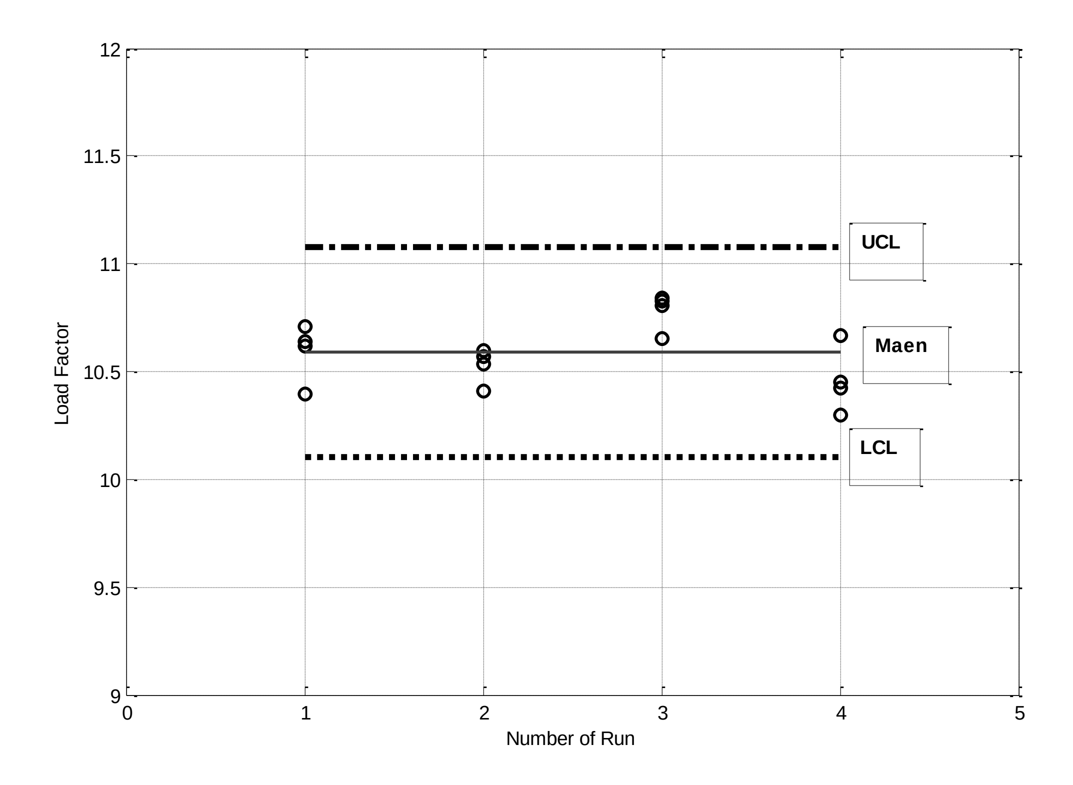


Figure A3: Xbar chart for point #3.

Table A4. Load factor for flight envelope point #1: 1.0 Mach & 5000 m Altitude.

Initialization Function	Population Size	Run1	Run2	Run3	Run4	**Mean**	**Standard Deviation**	**UCL**	**LCL**
		Nz	nz	nz	nz				
M. Gauss Function	64	10.657	10.529	10.571	10.509	**10.55**	**0.098**	**10.84**	**10.25**
M. Sinc Function	64	10.425	10.54	10.625	10.606				
M. Gauss Function	128	10.688	10.657	10.4816	10.511				
M. Sinc Function	128	10.614	10.459	10.311	10.563				

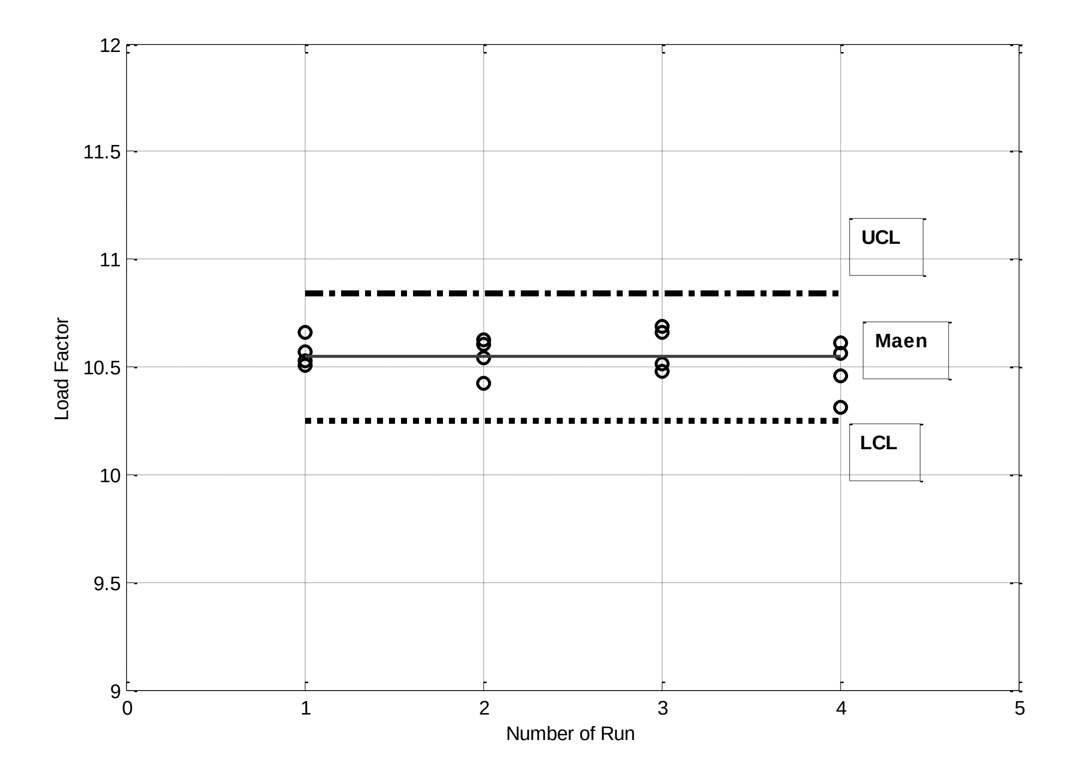


Figure A4: Xbar chart for point #4.

Table A5. Load factor for flight envelope point #1: 1.2 Mach & 3000 m Altitude.

Initialization Function	Population Size	Run1	Run2	Run3	Run4	Mean	Standard Deviation	UCL	LCL
		Nz	nz	nz	nz				
M. Gauss Function	64	10.425	10.449	10.664	10.301	**10.566**	**0.174**	**11.090**	**10.043**
M. Sinc Function	64	10.741	10.639	10.668	10.9				
M. Gauss Function	128	10.425	10.449	10.664	10.301				
M. Sinc Function	128	10.481	10.638	10.804	10.514				

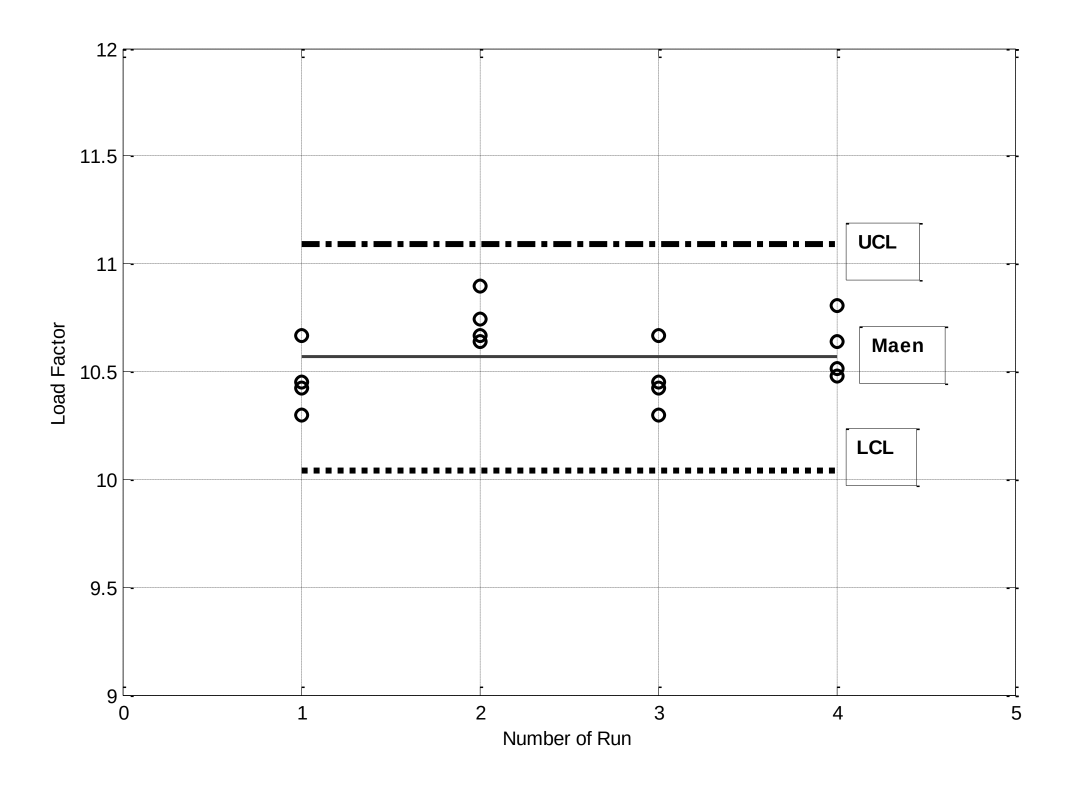


Figure A5: Xbar chart for point #5.

Table A6. Load factor for flight envelope point #1: 1.2 Mach & 6000 m Altitude.

Initialization Function	Population Size	Run1	Run2	Run3	Run4	**Mean**	**Standard Deviation**	**UCL**	**LCL**
		Nz	nz	nz	nz				
M. Gauss Function	64	10.484	10.546	10.461	10.485	**10.579**	**0.127**	**10.961**	**10.196**
M. Sinc Function	64	10.38	10.543	10.735	10.663				
M. Gauss Function	128	10.553	10.459	10.828	10.44				
M. Sinc Function	128	10.6	10.716	10.655	10.709				

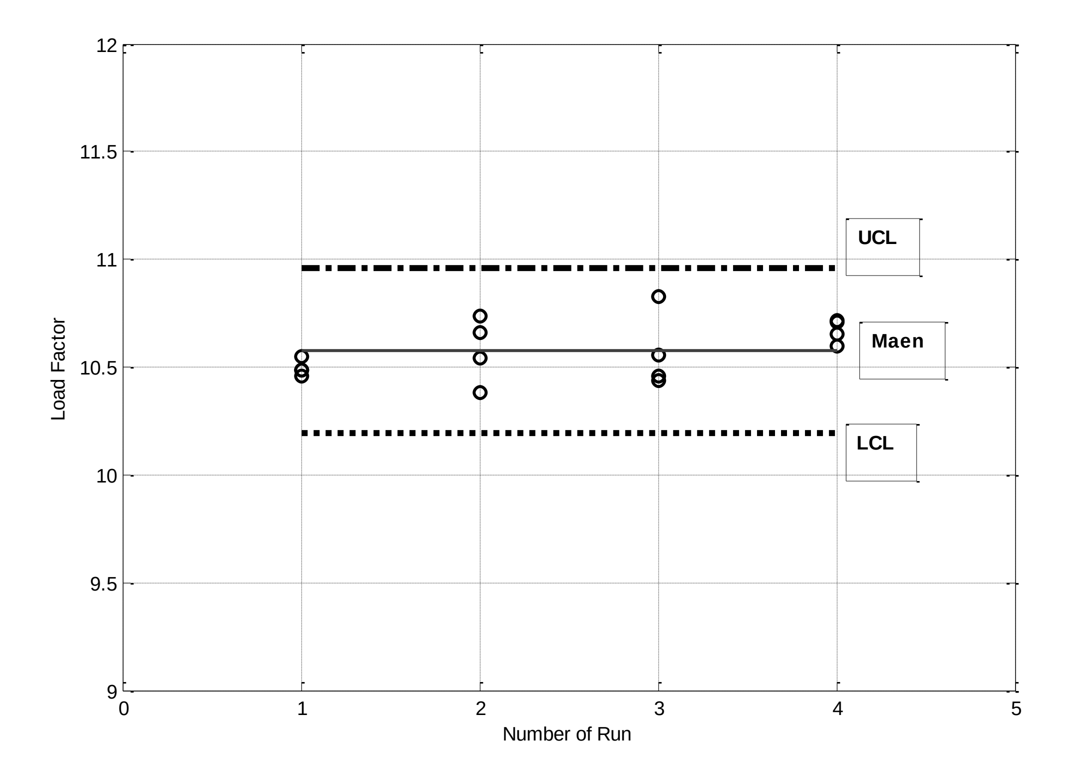


Figure A6: Xbar chart for point #6.

ABSTRACT IN ARABIC

استخدام الخوارزميات الجينية المتصلة من أجل ترخيص قوانين تحكم الطيران اللاخطية

إعداد
علي غالب علي العساسفه

المشرف
الأستاذ محمد نادر حمدان

المشرف المشارك
الدكتور زائر سالم ابوحمور

الملخص

استخدمت الخوارزميات الجينية الم تصله في هذه البحث من أجل حل مشكلة الأمثلة غير الخطية التي تنتج من عملية ترخيص قوانين تحكم الطيران اللاخطية. وتستخدم هذه الطريقة لتوليد أوامر الطيار التي تحكم أداء الطائرة حول نقاط معينة في مجال الطيران المسموح به. لقد تم اعتماد الأوامر التالية في هذه الدراسة: أوامر الطيار الخاصة بالتحكم بدرجة الميل و الانحراف حول المحور الصادي و أوامر الطيار الخاصة بالتحكم بدرجة الميل و الانحراف حول المحور السيني . ويتم تحليل أداء الطائرة الناتج عن هذه الأوامر من أجل العثور على أسوأ تركيبه قد تؤدي إلى تجاوزات في عامل الحمولة المسموح به أو زاوية الهجوم.

ويرجع الدافع لاستخدام الخوارزميات الجينية الم تصله لحل هذا النوع من المشاكل إلى حقيقة أن أوامر الطيار عادة ما تكون سلسة ومترابطة، والتي من الصعب توليدها باستخدام الخوارزميات الجينية التقليدية. الخوارزميات الجينية المتصله أيضا لديها ميزة أكثر من الأسلوب التقليدي للخوارزمية الجينية التقليدية بأن لها القدرة على توليد إشارات سلسة مناسبة دون فقدان بعض المعلومات و المميزات الهامة في تلك الإشارات, وكل ذلك في ظل وجود أجهزة تحديد معدل التغير في أنظمة التحكم في الطيران، و في ظل وجود زمن استجابة للمحركات ليس بقليل . وعلاوة على ذلك يمكن اعتبار الخوارزميات الجينية الم تصله طريقه مناسبة لاستخدامها لإيجاد الحلول المثلى الشاملة، ويمكنها التعامل مع الأنظمة متعددة المداخل.

وتعرض نتائج المحاكاة الأداء المتفوق النسبي لاستخدام الخوارزميات الجينية المتصله مقارنة مع الخوارزميات الوراثية التقليدية.

الكلمات الدليلية: مشاكل السيطرةِ المثاليةِ، الخوارزميات الجينية المستمرة, الأنظمة اللاخطية للتحكم الطيران